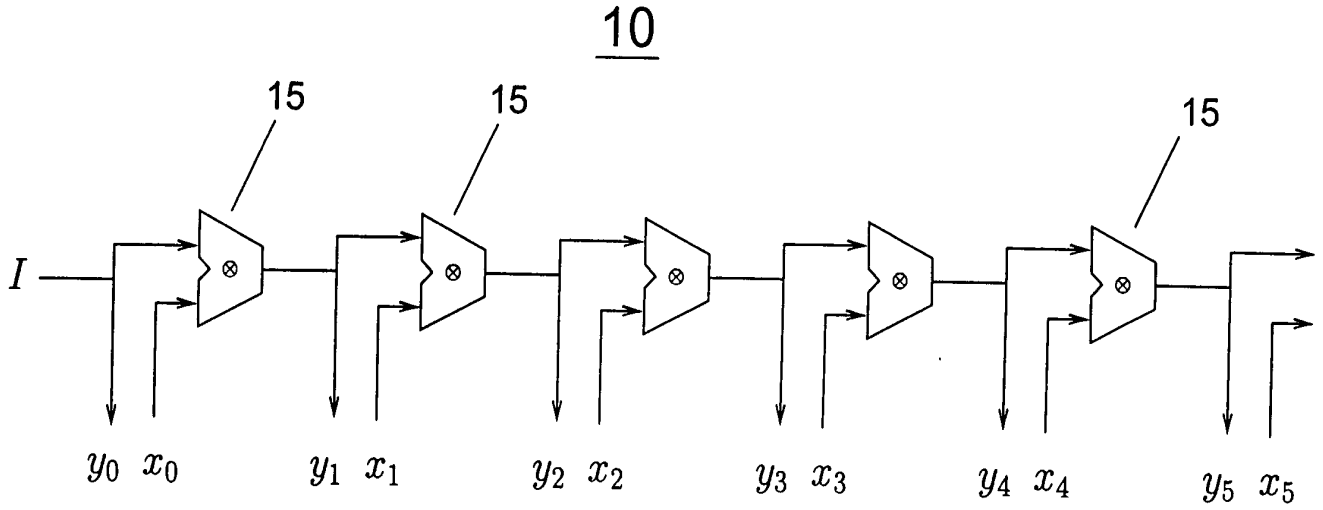


FIG. 1 (PRIOR ART)

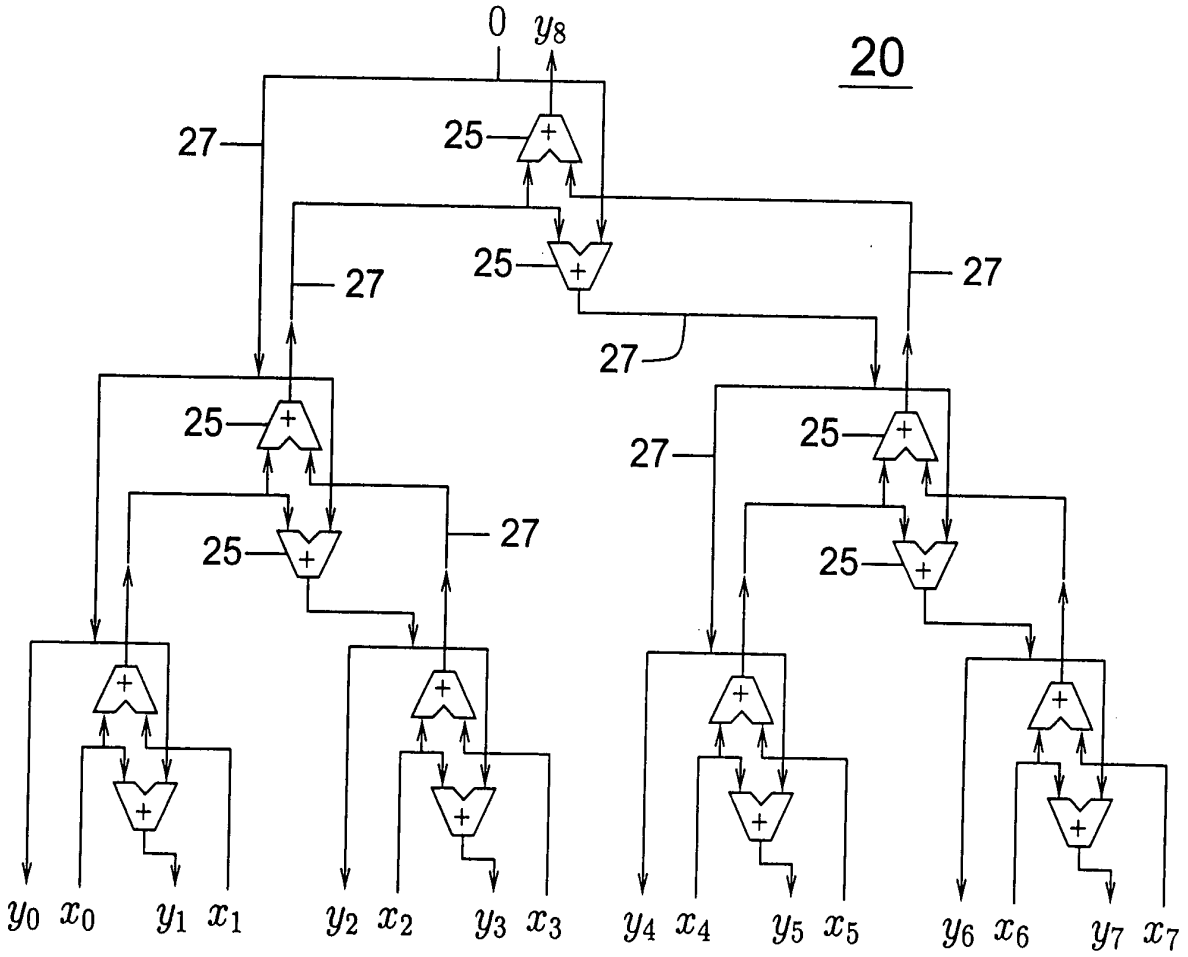


FIG. 2 (PRIOR ART)

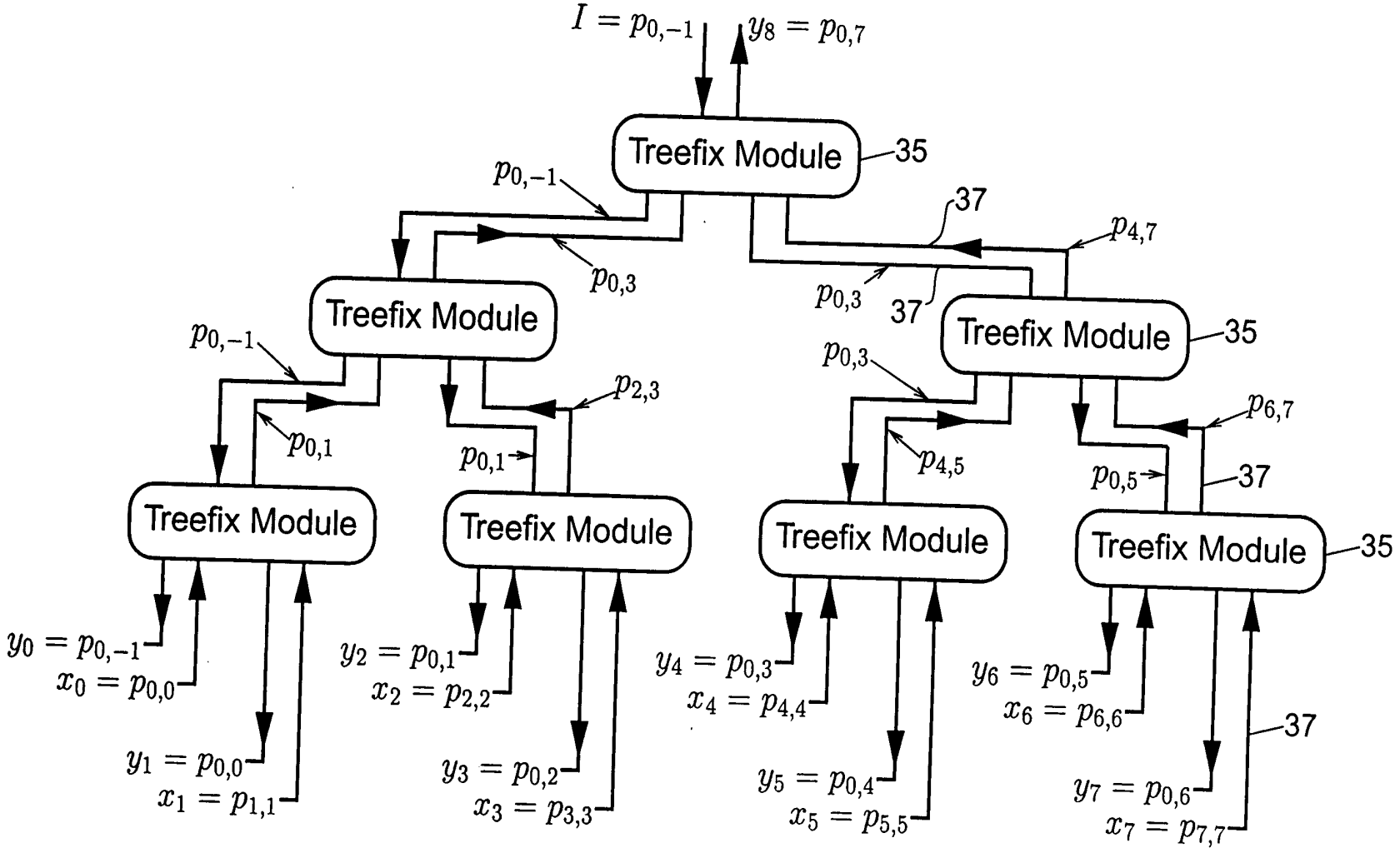
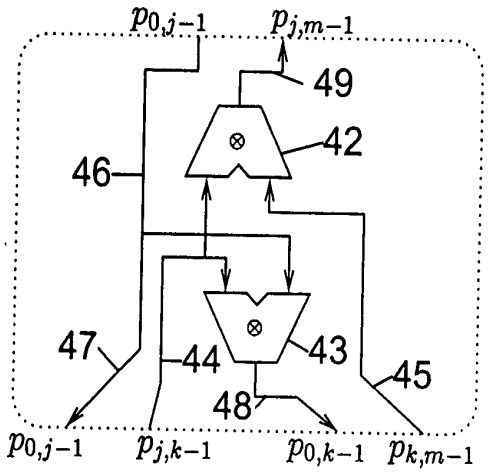


FIG. 3 (PRIOR ART)



35

FIG. 4 (PRIOR ART)

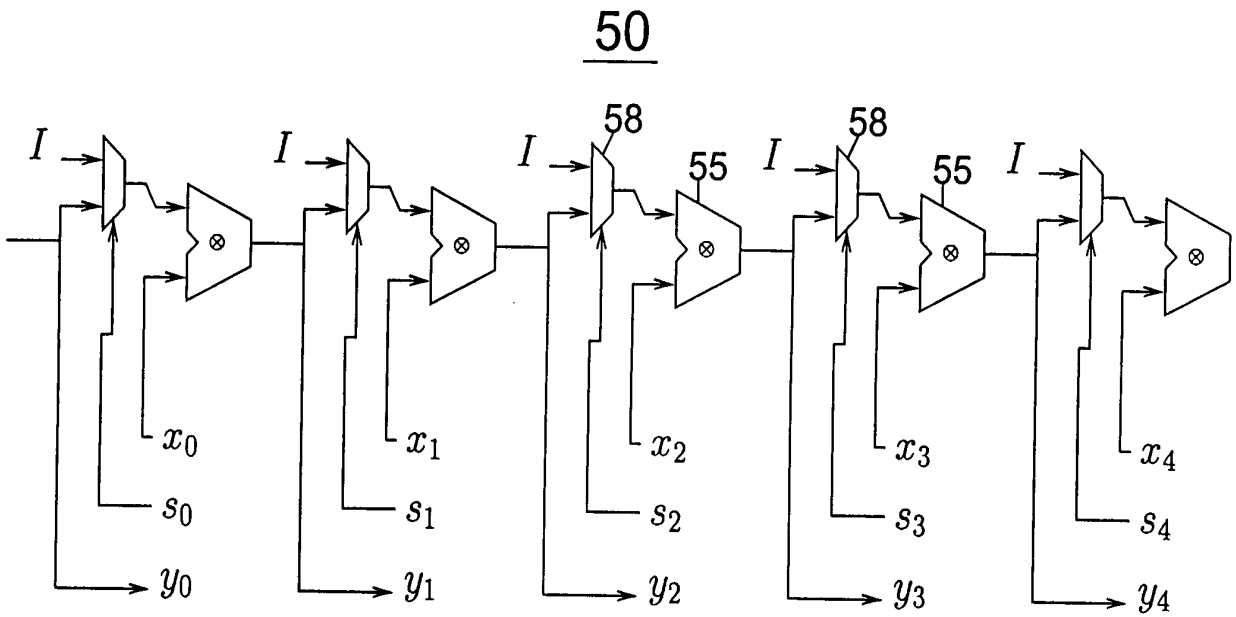


FIG. 5 (PRIOR ART)

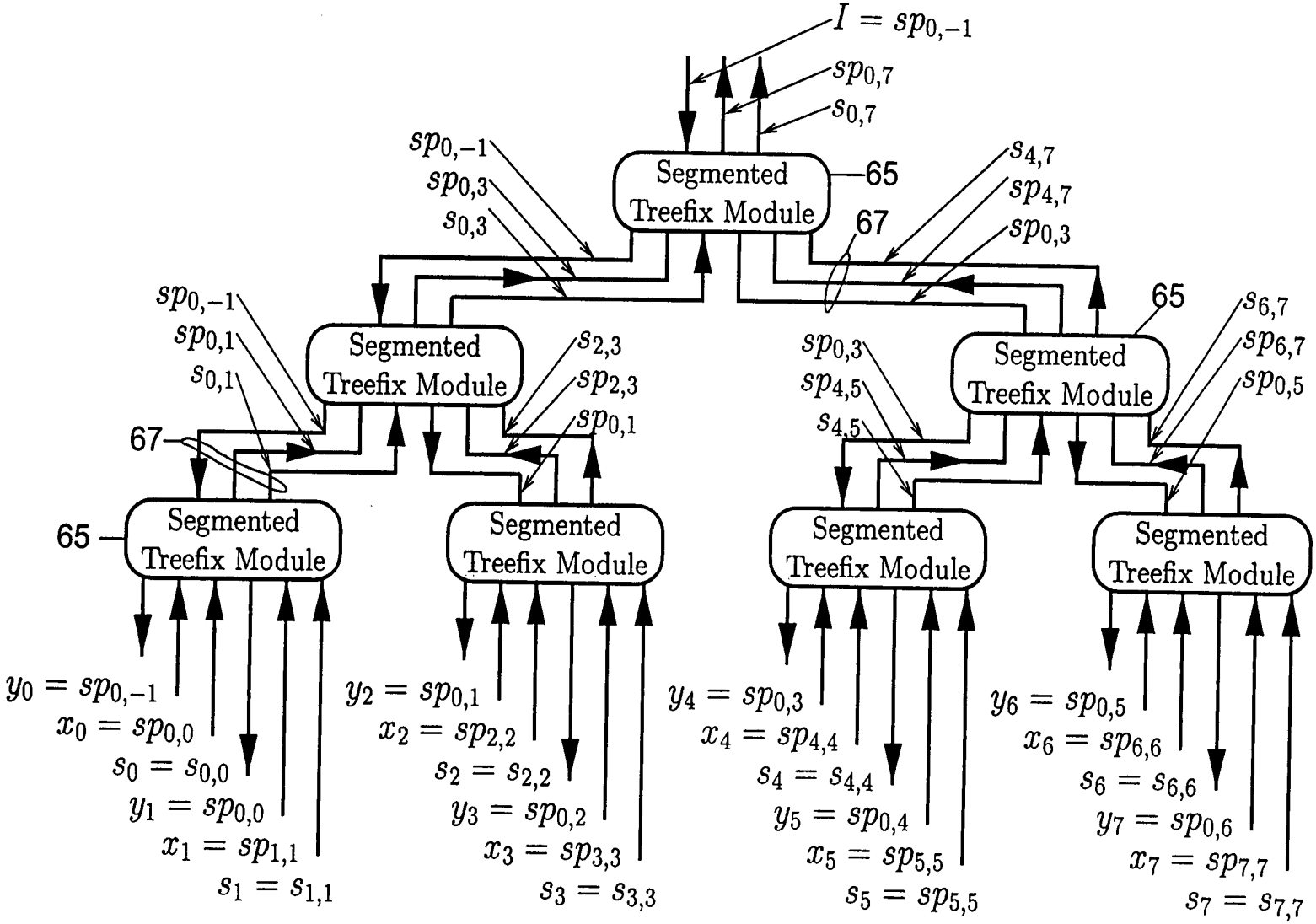


FIG. 6 (PRIOR ART)

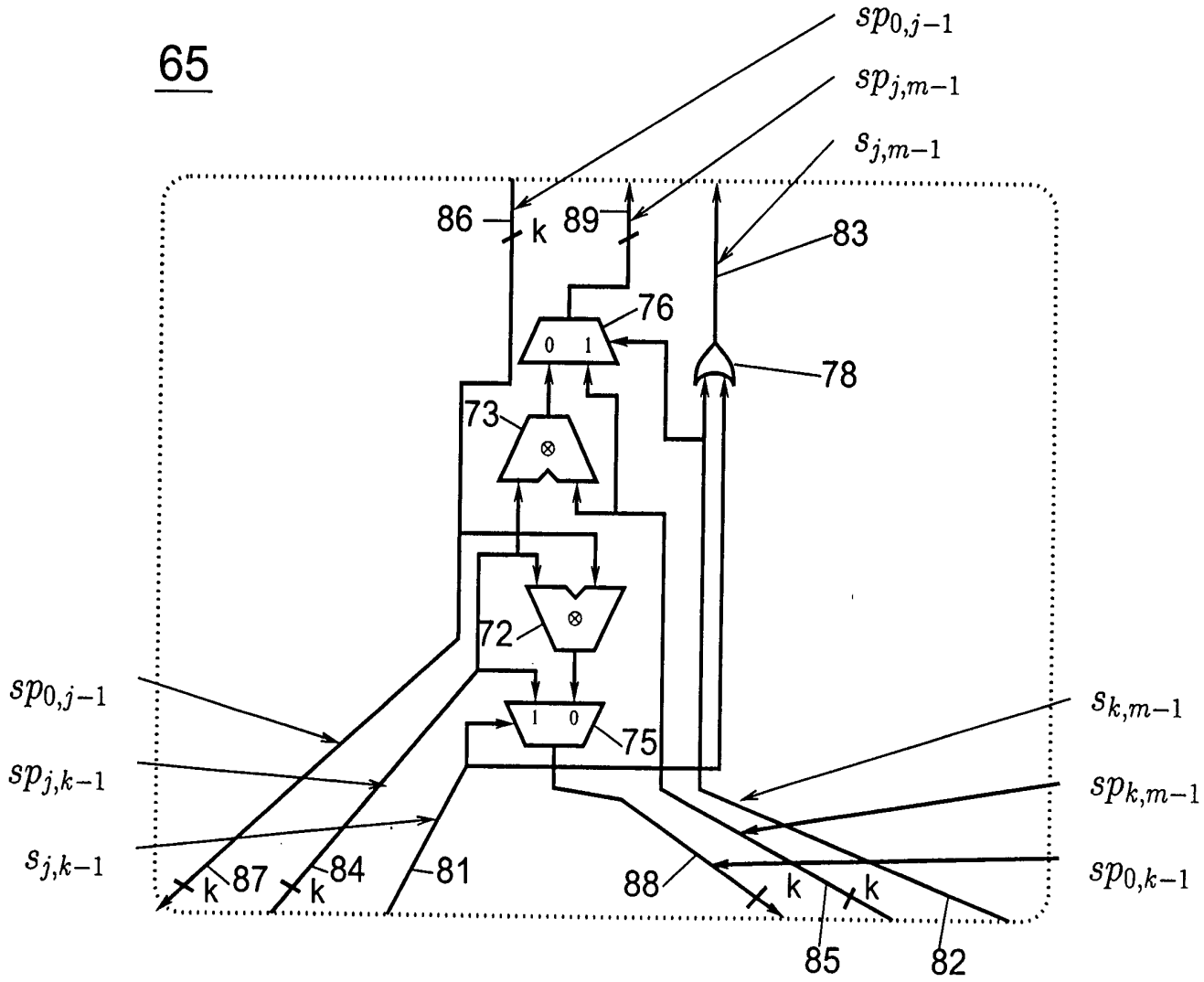


FIG. 7 (PRIOR ART)

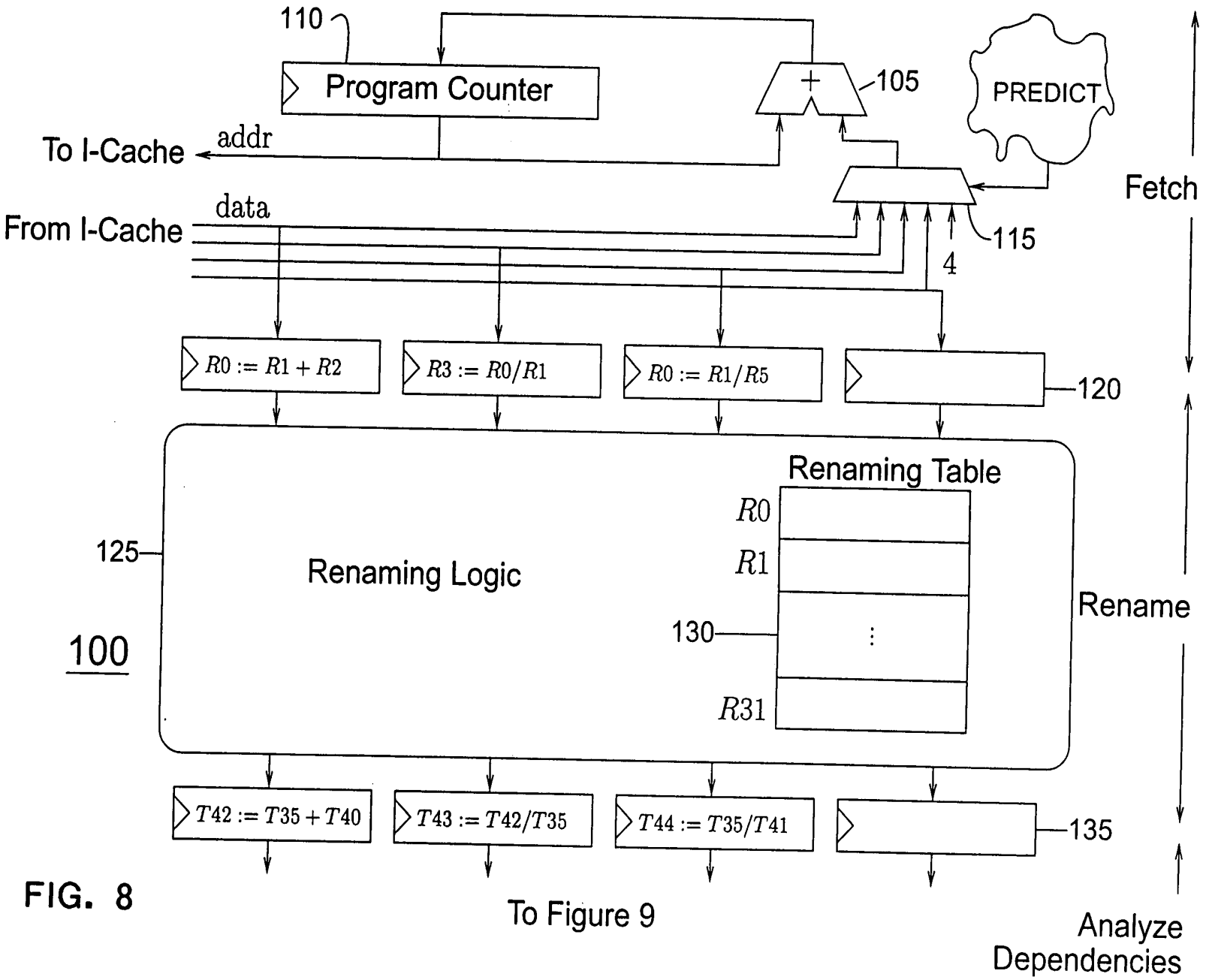

110
Program Counter
105
PREDICT
To I-Cache
addr
From I-Cache
data
115
4
Fetch
R0 := R1 + R2
R3 := R0/R1
R0 := R1/R5
120
Renaming Table
R0
R1
130
R31
125
Renaming Logic
100
Rename
T42 := T35 + T40
T43 := T42/T35
T44 := T35/T41
135
To Figure 9
Analyze Dependencies

FIG. 8

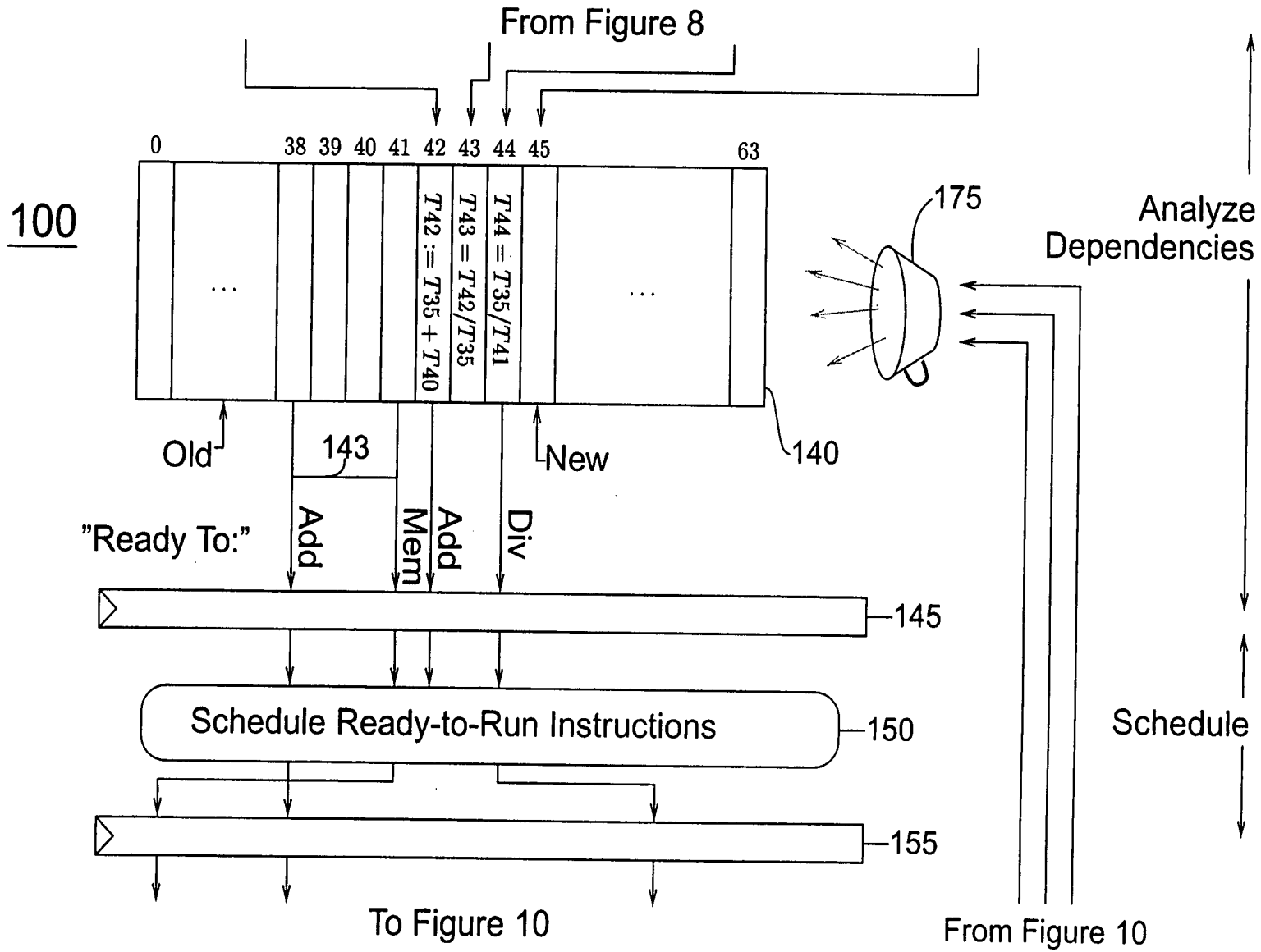


FIG. 9 (PRIOR ART)

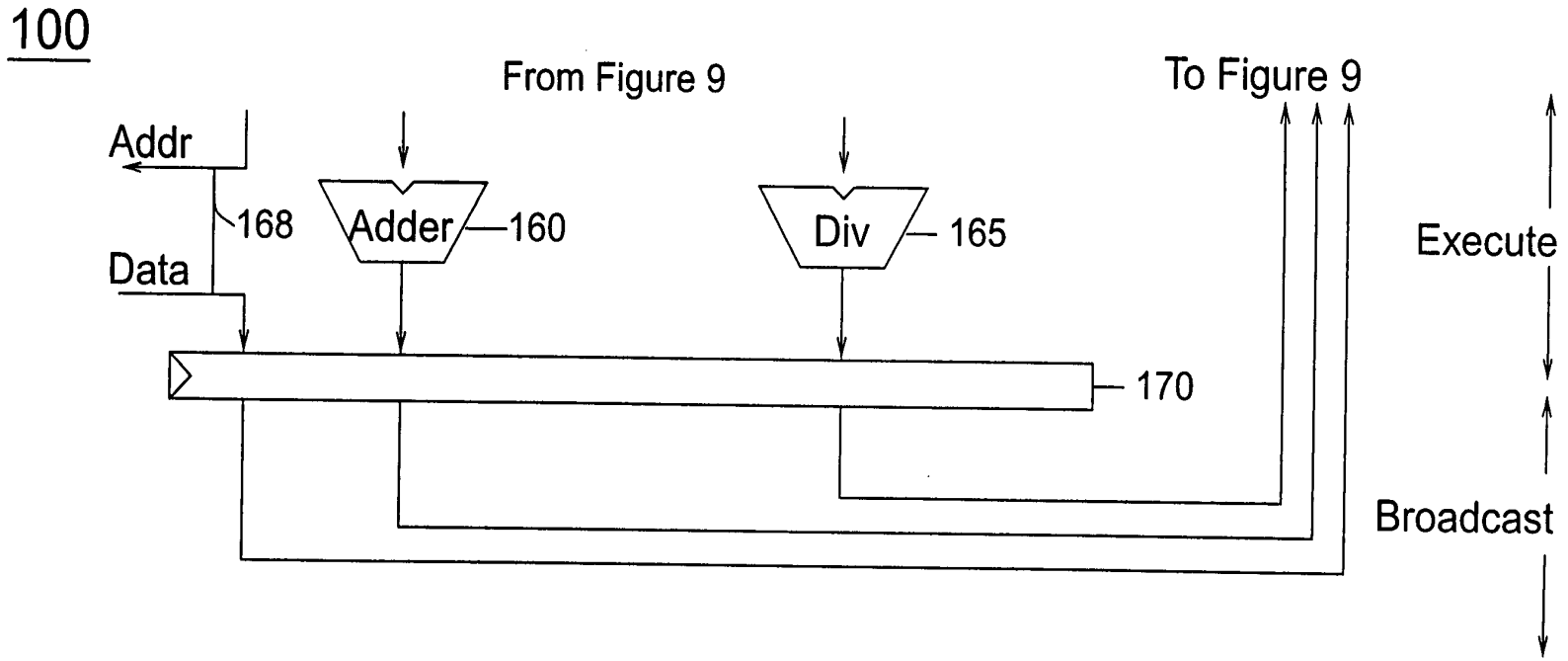


FIG. 10 (PRIOR ART)

Register	Tag
0	30
1	35
2	40
3	25
4	26
5	41

Before

Register	Tag
0	42
1	35
2	40
3	25
4	26
5	41

After Instruction 0

Register	Tag
0	42
1	35
2	40
3	43
4	26
5	41

After Instruction 1

Register	Tag
0	44
1	35
2	40
3	43
4	26
5	41

After Instruction 2

FIG. 11

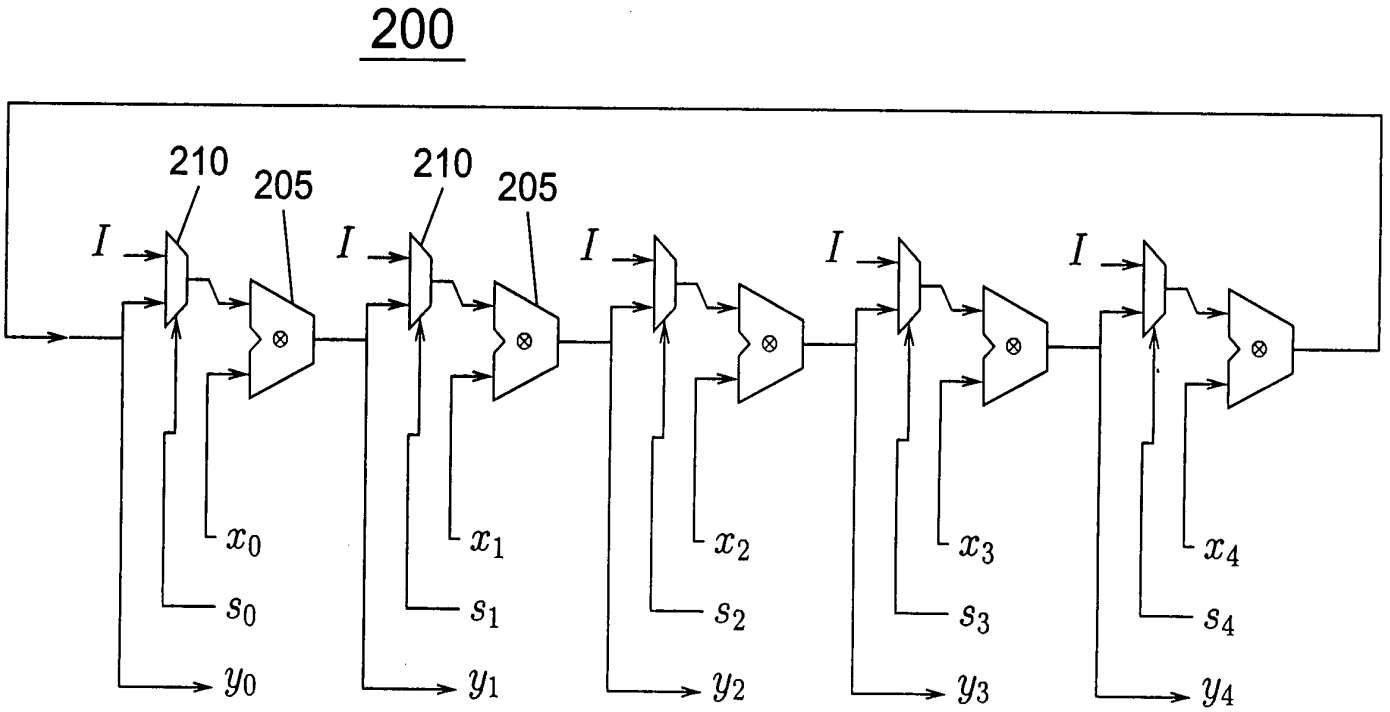

200
210
205
210
205
I
I
I
I
I
x_0
x_1
x_2
x_3
x_4
s_0
s_1
s_2
s_3
s_4
y_0
y_1
y_2
y_3
y_4

FIG. 12

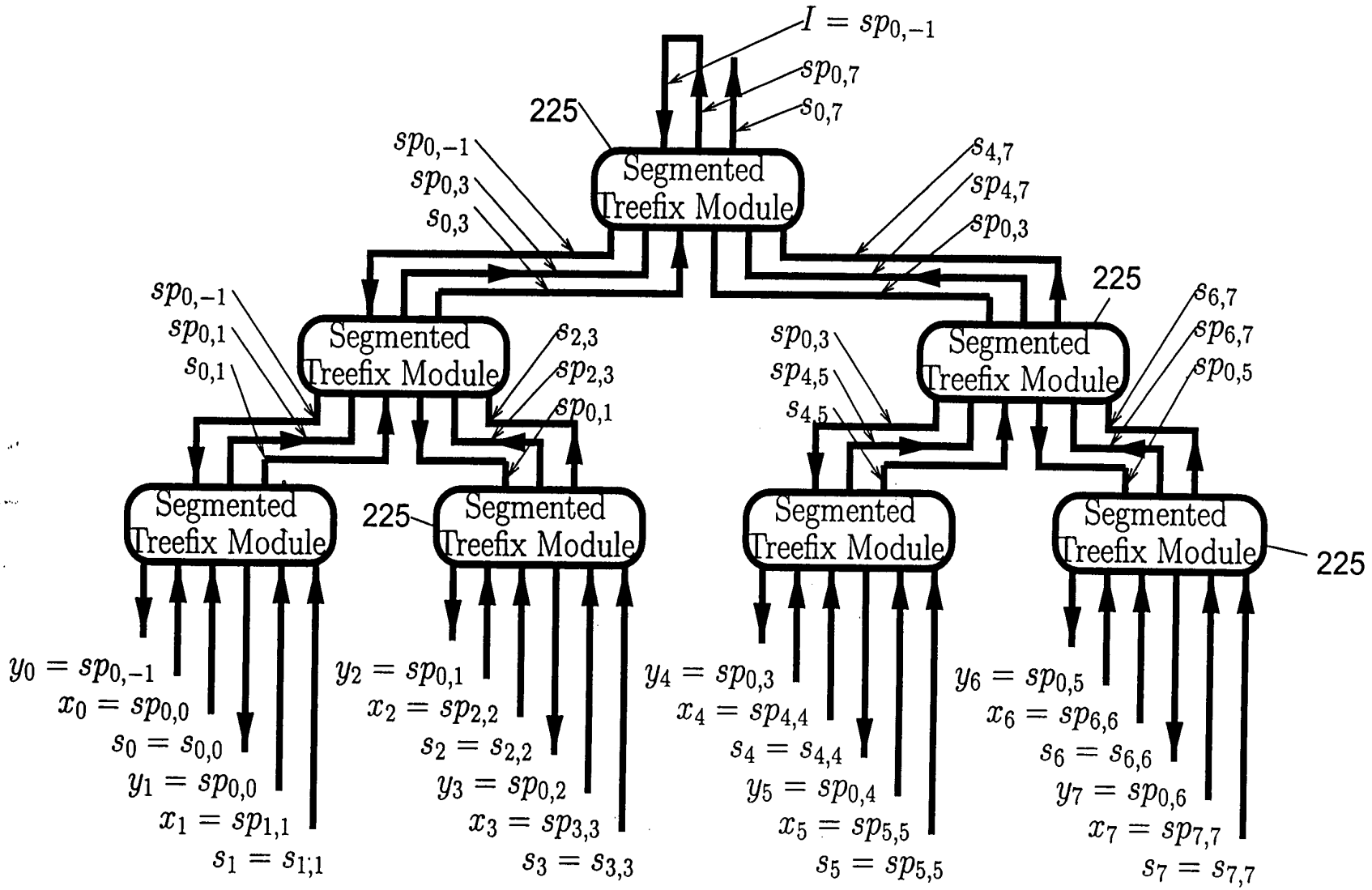

$I = sp_{0,-1}$
$sp_{0,7}$
$s_{0,7}$
225
$sp_{0,-1}$
$sp_{0,3}$
$s_{0,3}$
Segmented Treefix Module
$s_{4,7}$
$sp_{4,7}$
$sp_{0,3}$
225
$sp_{0,-1}$
$sp_{0,1}$
$s_{0,1}$
Segmented Treefix Module
$s_{2,3}$
$sp_{2,3}$
$sp_{0,1}$
$sp_{0,3}$
$sp_{4,5}$
$s_{4,5}$
Segmented Treefix Module
$s_{6,7}$
$sp_{6,7}$
$sp_{0,5}$
Segmented Treefix Module
225
Segmented Treefix Module
Segmented Treefix Module
Segmented Treefix Module
Segmented Treefix Module
225
$y_0 = sp_{0,-1}$
$x_0 = sp_{0,0}$
$s_0 = s_{0,0}$
$y_1 = sp_{0,0}$
$x_1 = sp_{1,1}$
$s_1 = s_{1;1}$
$y_2 = sp_{0,1}$
$x_2 = sp_{2,2}$
$s_2 = s_{2,2}$
$y_3 = sp_{0,2}$
$x_3 = sp_{3,3}$
$s_3 = s_{3,3}$
$y_4 = sp_{0,3}$
$x_4 = sp_{4,4}$
$s_4 = s_{4,4}$
$y_5 = sp_{0,4}$
$x_5 = sp_{5,5}$
$s_5 = sp_{5,5}$
$y_6 = sp_{0,5}$
$x_6 = sp_{6,6}$
$s_6 = s_{6,6}$
$y_7 = sp_{0,6}$
$x_7 = sp_{7,7}$
$s_7 = s_{7,7}$

FIG. 13

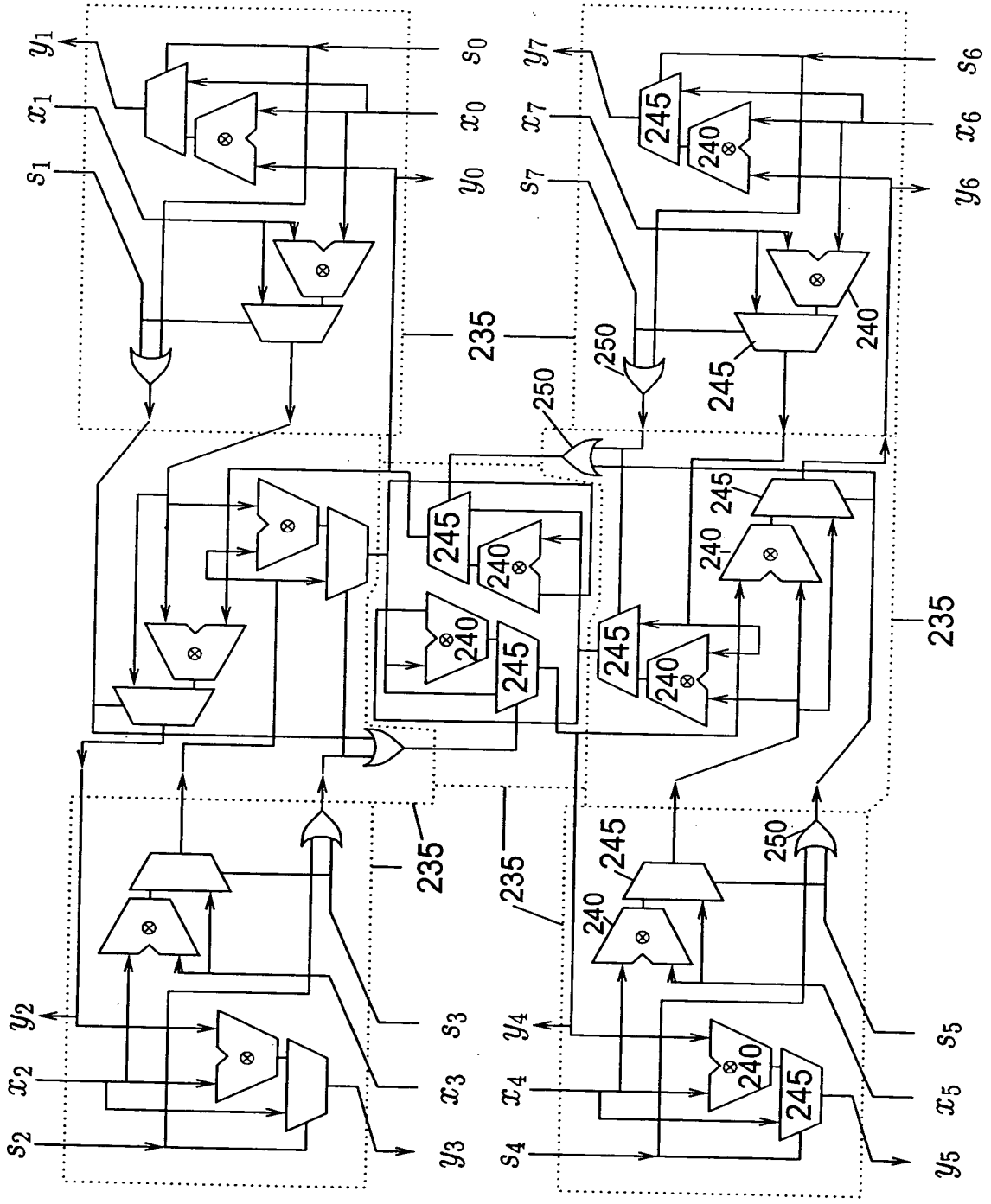

y_1
x_1
s_1
s_0
x_0
y_0
y_7
x_7
s_7
s_6
x_6
y_6
y_2
x_2
s_2
s_3
x_3
y_3
y_4
x_4
s_4
s_5
x_5
y_5
235
240
245
250

FIG. 14

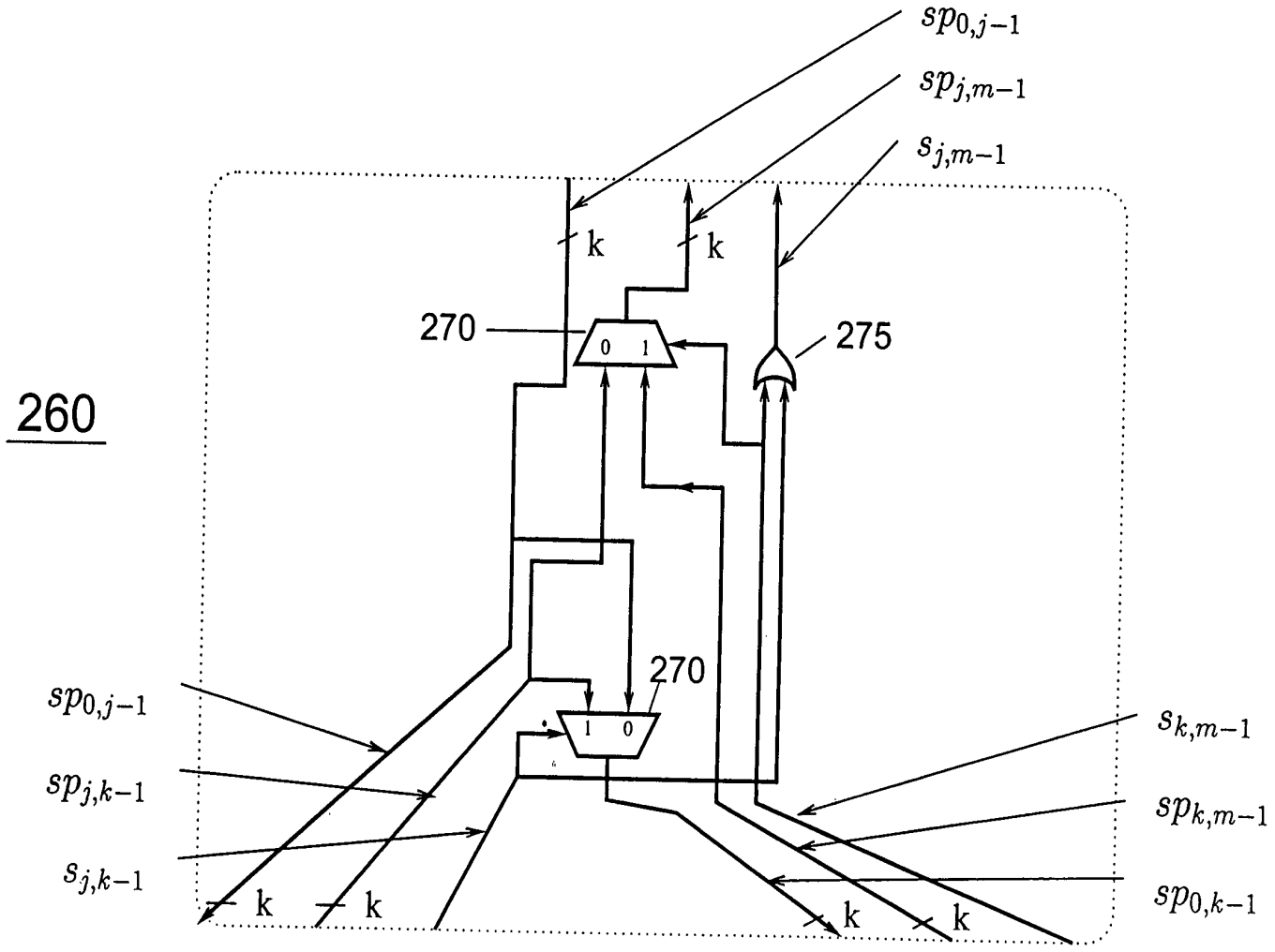

260
$sp_{0,j-1}$
$sp_{j,m-1}$
$s_{j,m-1}$
k
k
270
0 1
275
270
1 0
$sp_{0,j-1}$
$sp_{j,k-1}$
$s_{j,k-1}$
k
k
$s_{k,m-1}$
$sp_{k,m-1}$
$sp_{0,k-1}$
k
k

FIG. 15

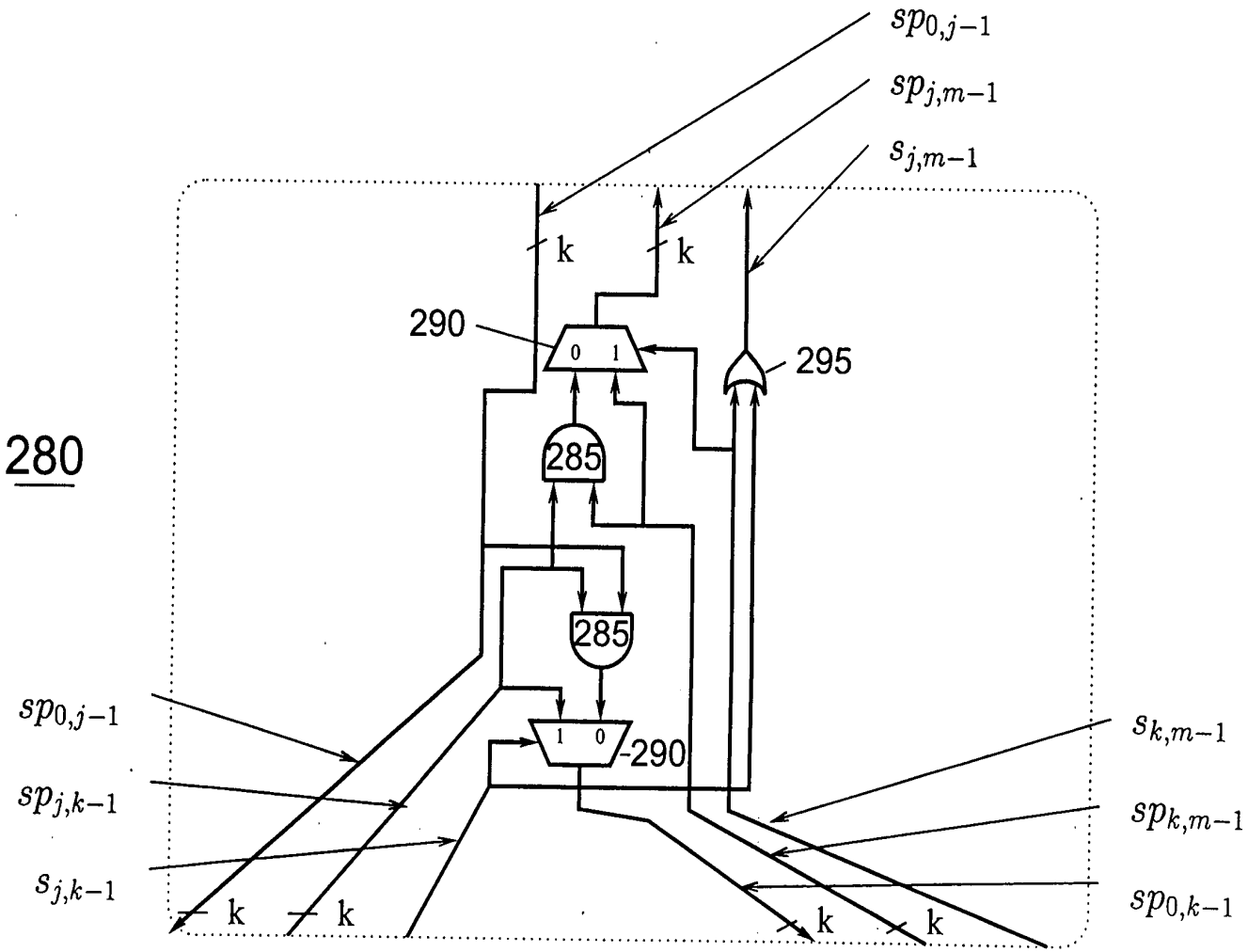

$sp_{0,j-1}$
$sp_{j,m-1}$
$s_{j,m-1}$
k
k
290
0 1
295
280
285
285
$sp_{0,j-1}$
$sp_{j,k-1}$
$s_{j,k-1}$
1 0
290
$s_{k,m-1}$
$sp_{k,m-1}$
$sp_{0,k-1}$
k
k
k
k

FIG. 16

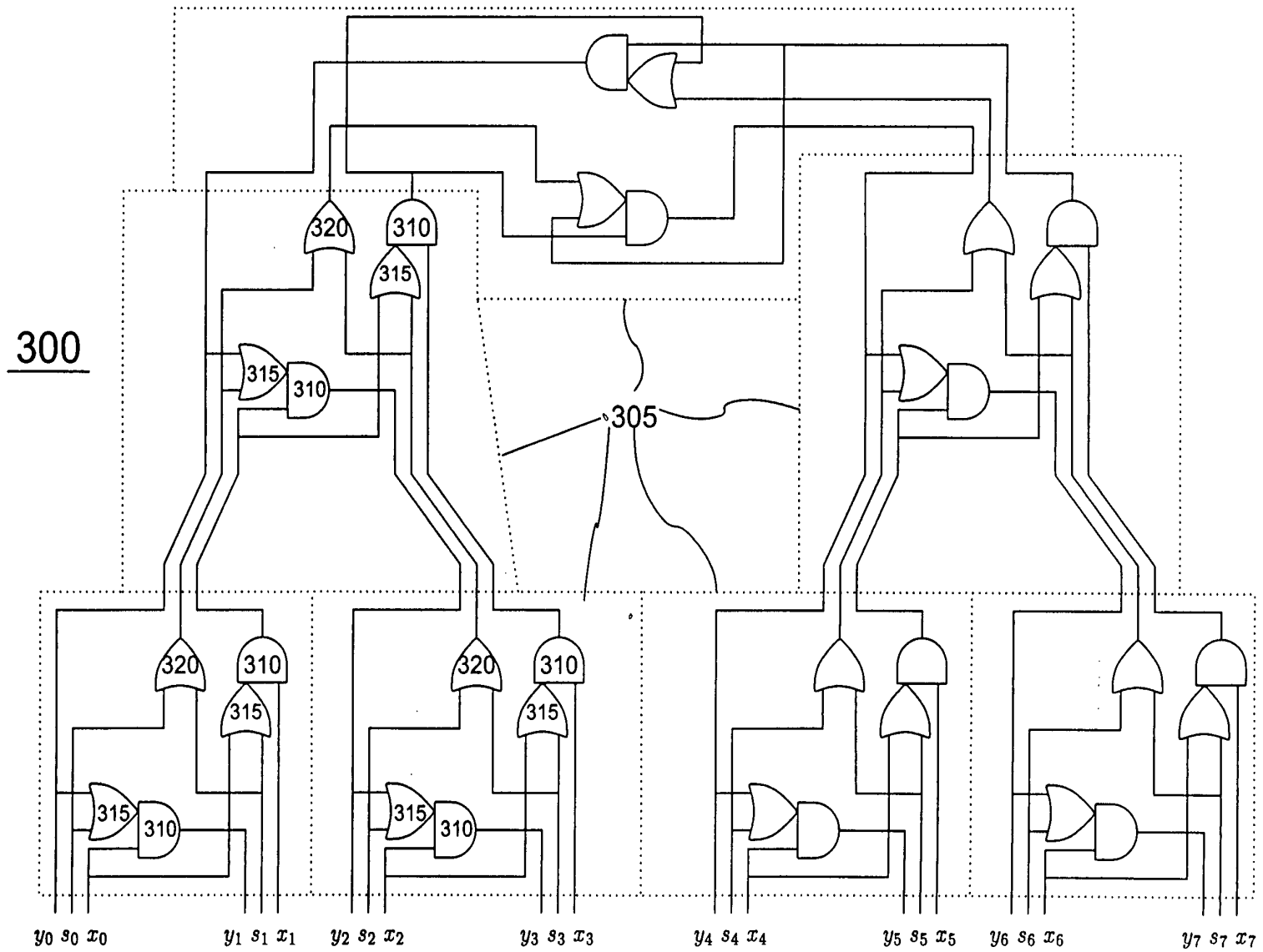

300
305
320
310
315
y_0 s_0 x_0
y_1 s_1 x_1
y_2 s_2 x_2
y_3 s_3 x_3
y_4 s_4 x_4
y_5 s_5 x_5
y_6 s_6 x_6
y_7 s_7 x_7

FIG. 17

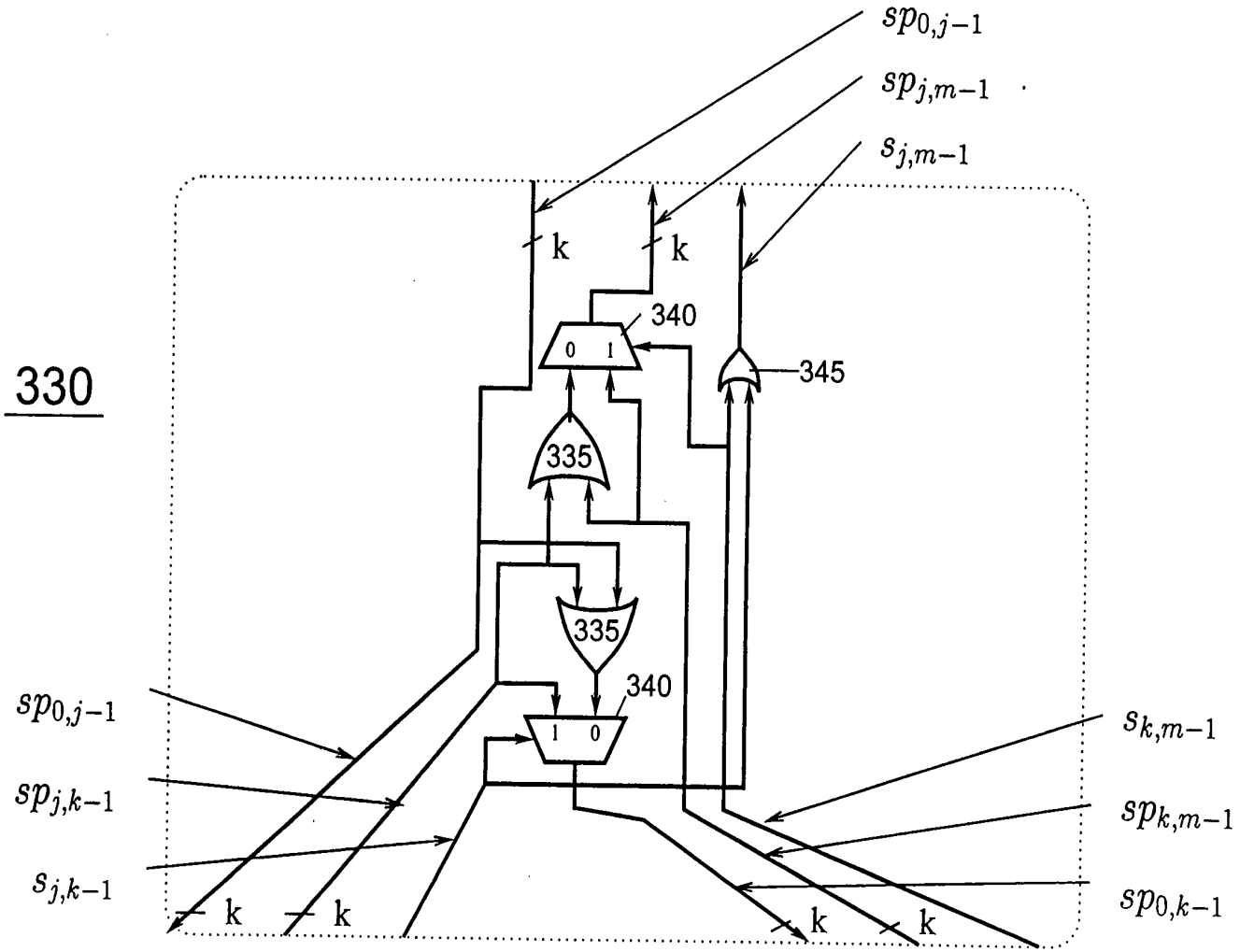

330
$sp_{0,j-1}$
$sp_{j,m-1}$
$s_{j,m-1}$
k
k
340
0 1
345
335
335
340
1 0
$sp_{0,j-1}$
$sp_{j,k-1}$
$s_{j,k-1}$
$s_{k,m-1}$
$sp_{k,m-1}$
$sp_{0,k-1}$
k
k
k
k

FIG. 18

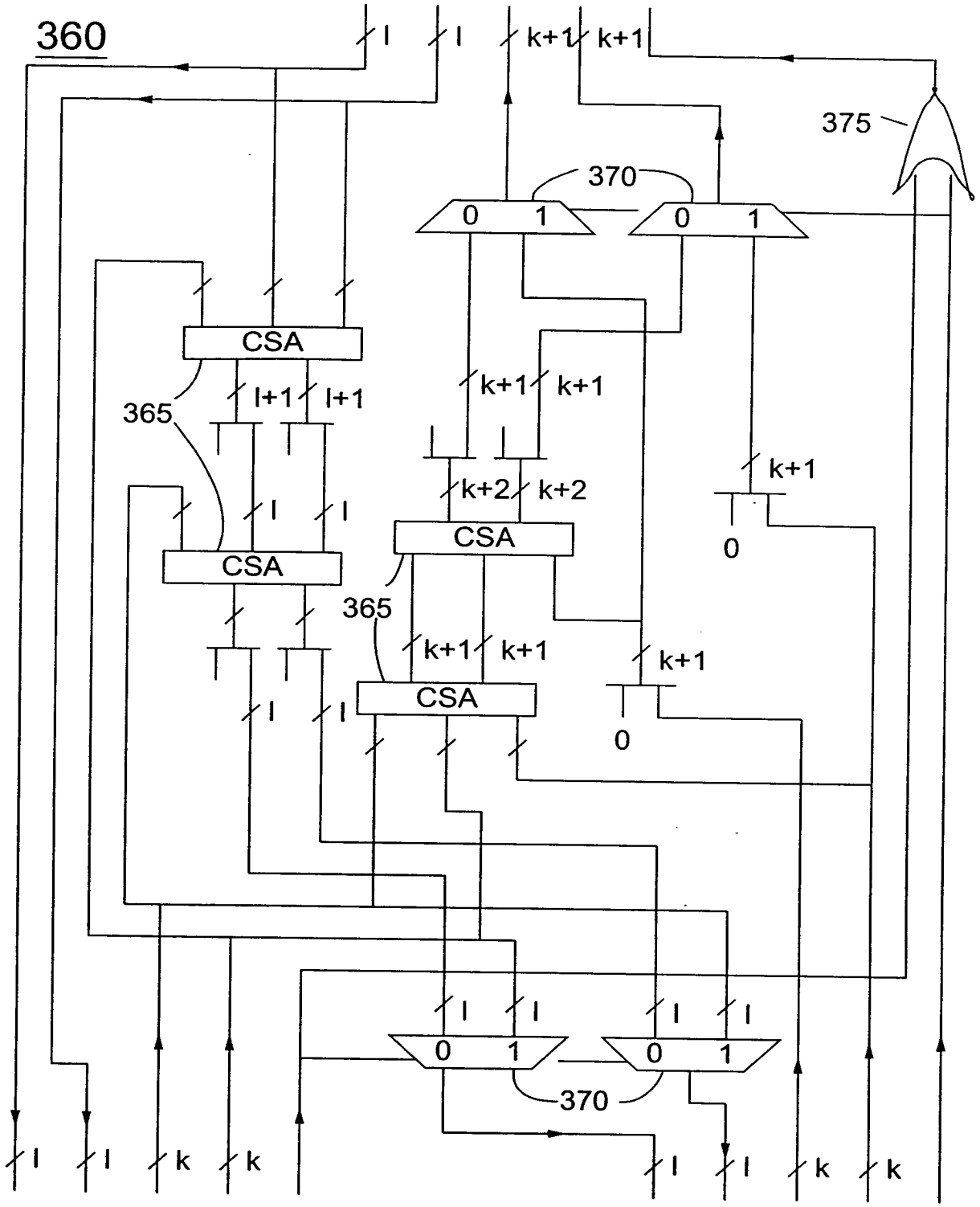

360
l
l
k+1
k+1
375
370
0
1
0
1
CSA
365
l+1
l+1
k+1
k+1
k+1
k+2
k+2
l
l
CSA
CSA
0
365
k+1
k+1
k+1
l
l
CSA
0
l
l
l
l
0
1
0
1
370
l
l
k
k
l
l
k
k

FIG. 19

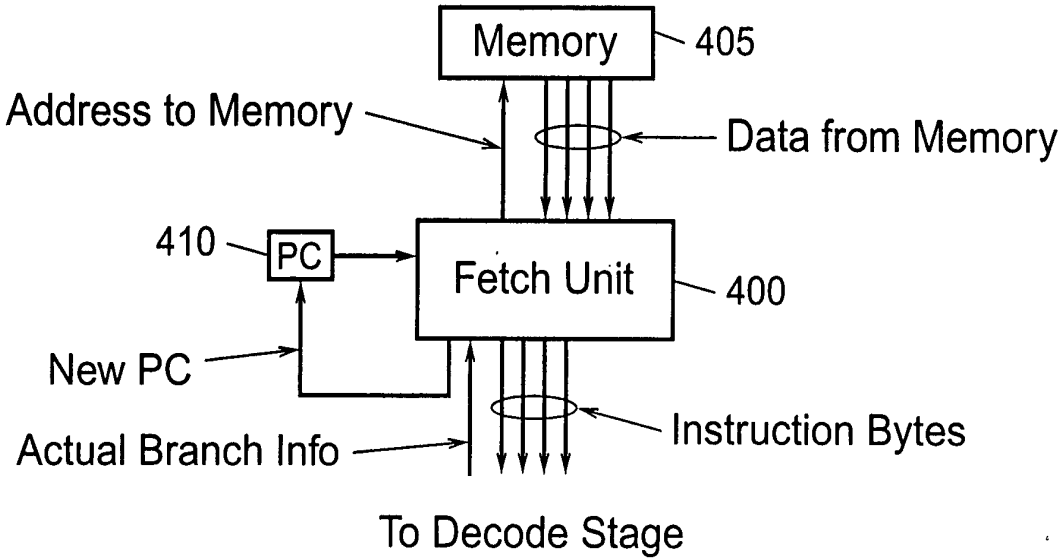


FIG. 20 (PRIOR ART)

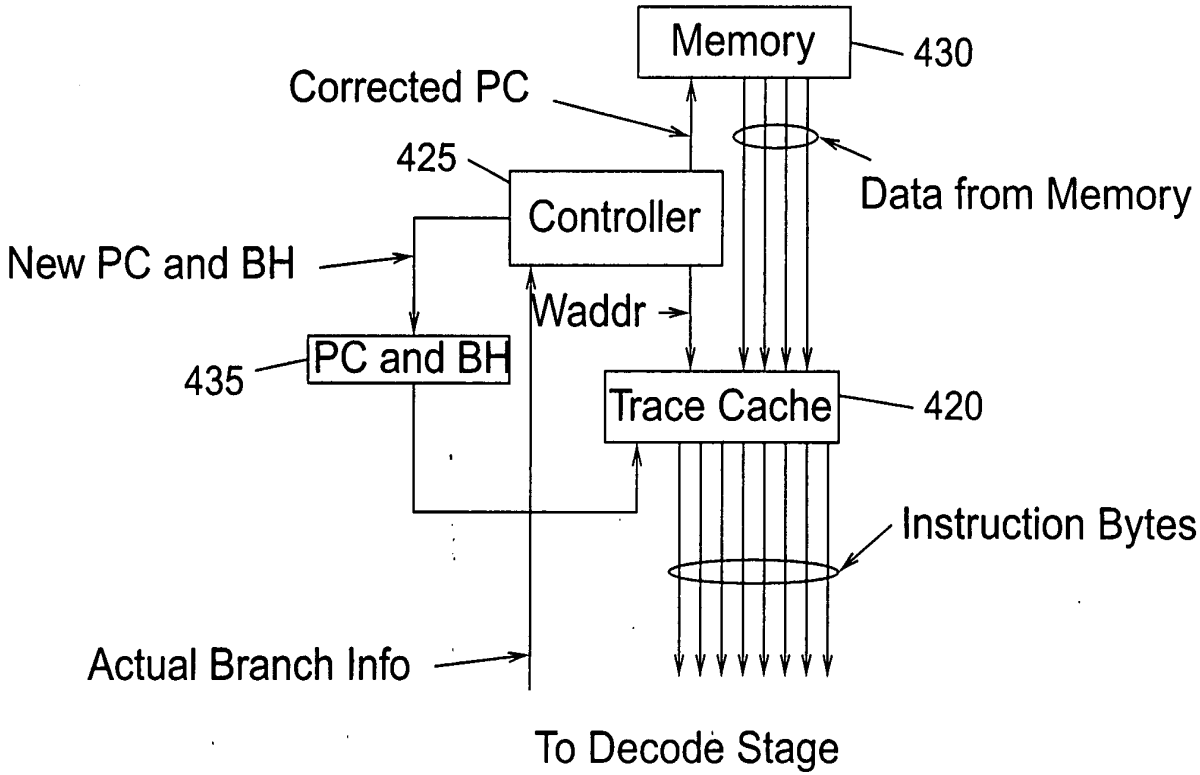


FIG. 21 (PRIOR ART)

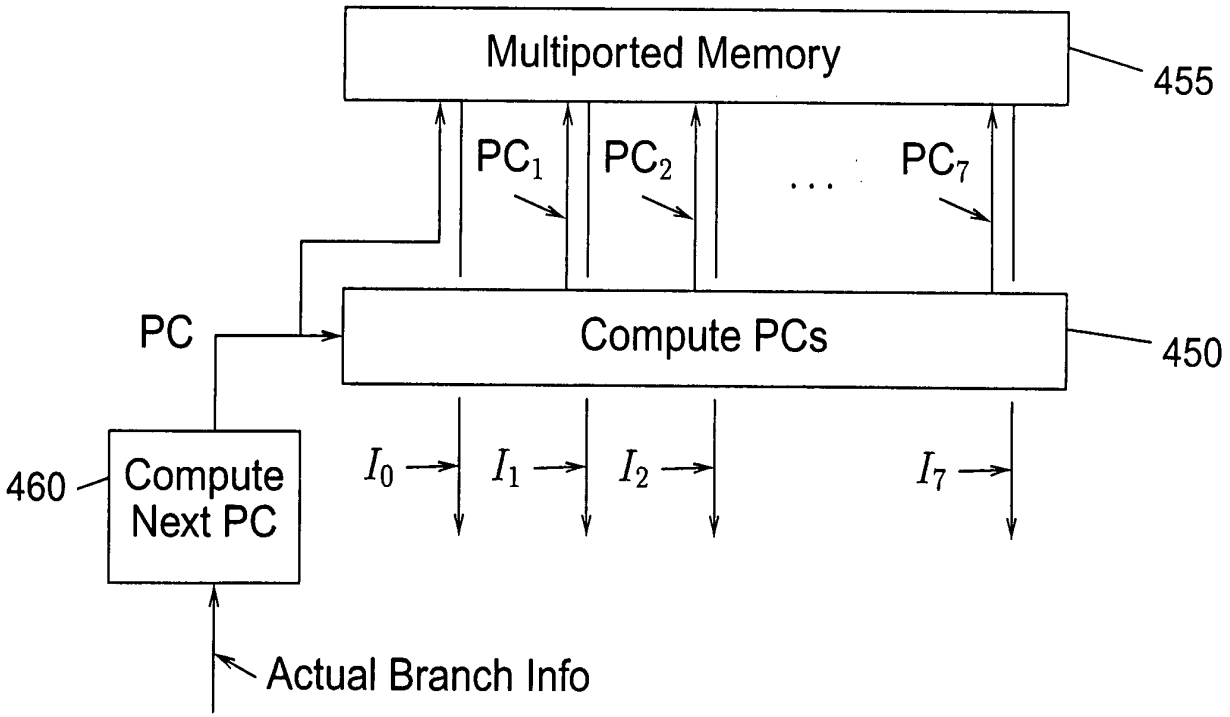

Multiported Memory
455
PC1
PC2
...
PC7
PC
Compute PCs
450
460
Compute Next PC
I0
I1
I2
I7
Actual Branch Info
To Decode Stage

FIG. 22

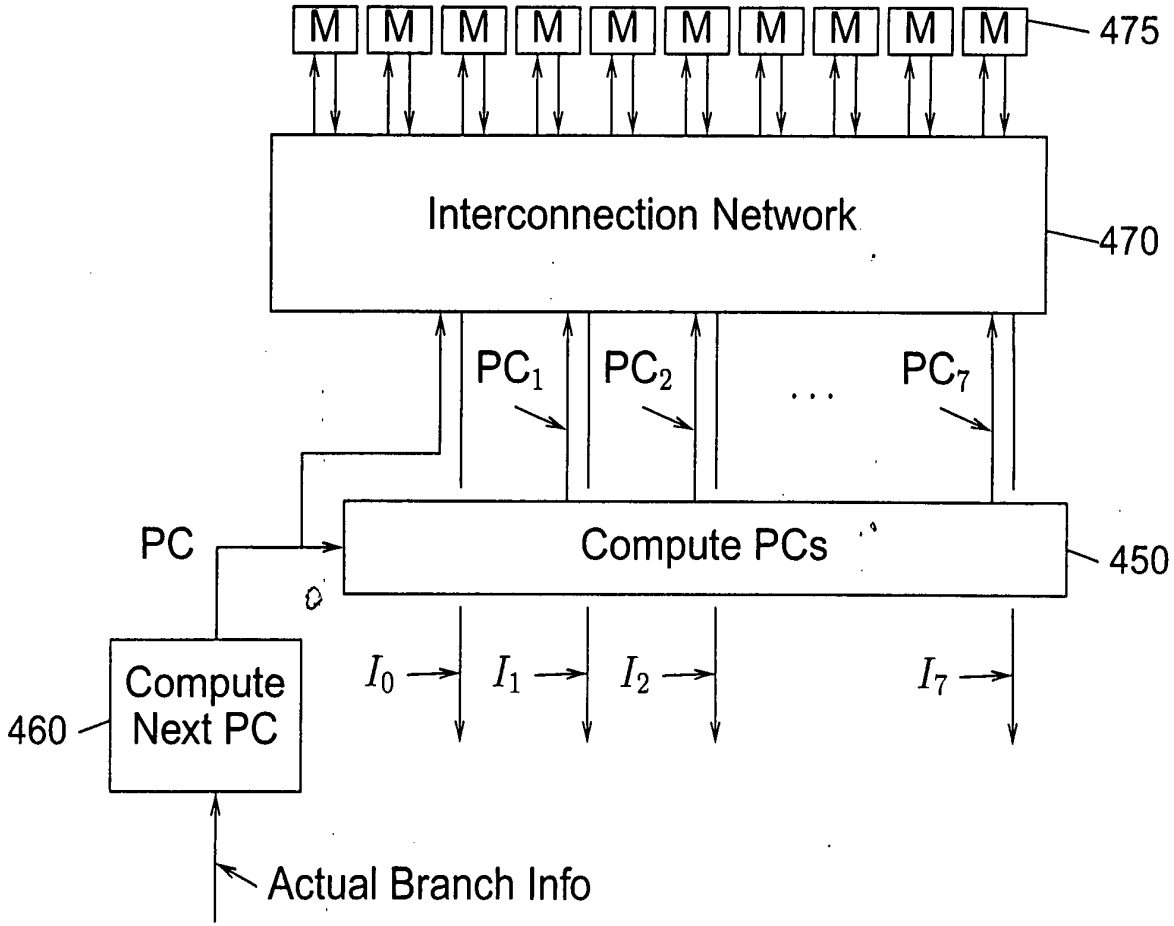

M
475
Interconnection Network
470
PC1
PC2
PC7
PC
Compute PCs
450
I0
I1
I2
I7
Compute
Next PC
460
Actual Branch Info
To Decode Stage

FIG. 23

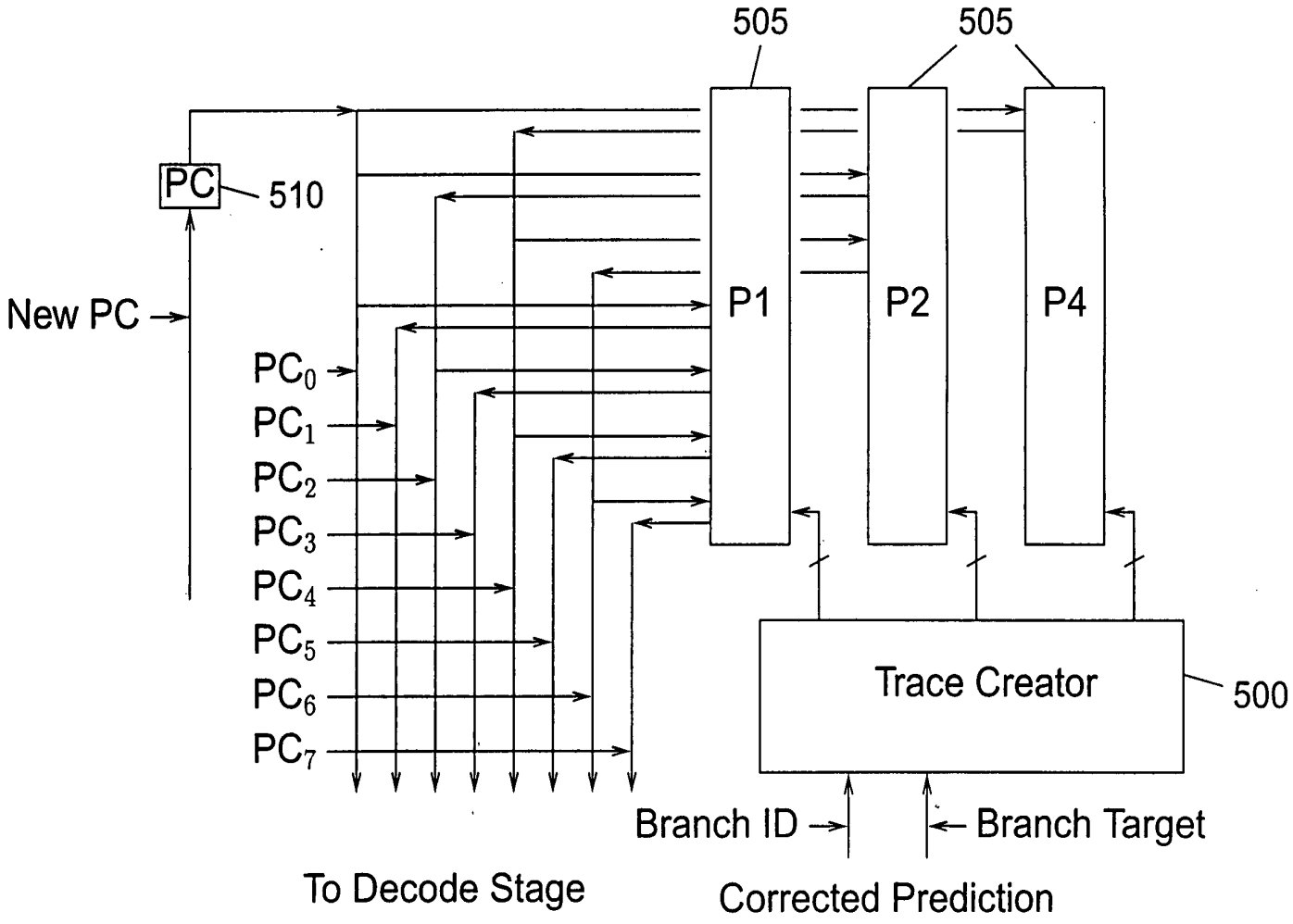
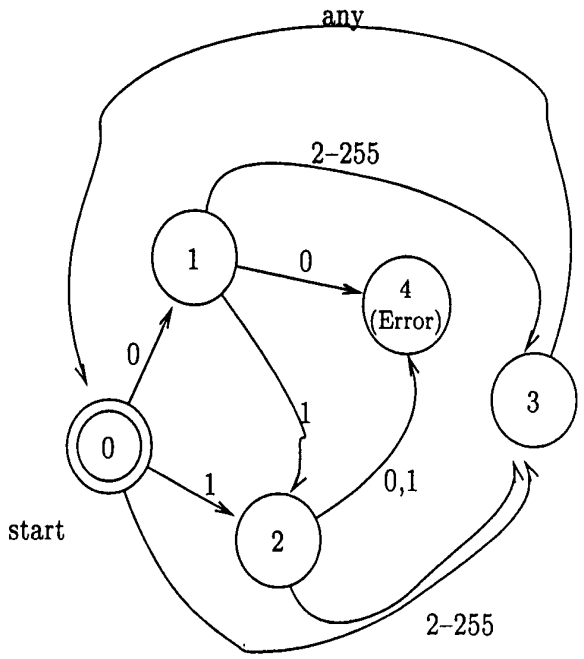

505
505
PC
510
New PC
P1
P2
P4
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Trace Creator
500
Branch ID
Branch Target
To Decode Stage
Corrected Prediction

FIG. 24



(a) State transition diagram.

State	input 0	1	2–255
0	1	2	3
1	4	2	3
2	4	4	3
3	0	0	0
4	4	4	4

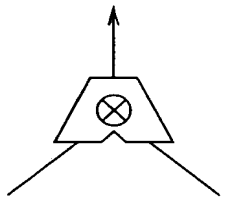
(b) State transistion table.

FIG. 25

Input	0	1	3	0	1	3	0	1
Level 0	1	2	3	1	2	3	1	2
	4	2	3	4	2	3	4	2
	4	4	3	4	4	3	4	4
	0	0	0	0	0	0	0	0
	4	4	4	4	4	4	4	4
Level 1	2		0		3		2	
	4		0		3		4	
	4		0		4		4	
	[2]		0		3		2	
	4		0		4		4	
Level 2		0				2		
		4				2		
		4				4		
		0				2		
		4				4		
Level 3				2				
				4				
				4				
				2				
				4				

FIG. 26

$$(x \otimes y) = \begin{bmatrix} (x \otimes y)_0 \\ (x \otimes y)_1 \\ (x \otimes y)_2 \\ (x \otimes y)_3 \\ (x \otimes y)_4 \end{bmatrix} = \begin{bmatrix} y_{x_0} \\ y_{x_1} \\ y_{x_2} \\ y_{x_3} \\ y_{x_4} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_4 \\ y_4 \\ y_0 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 4 \\ 2 \\ 4 \end{bmatrix}$$



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 4 \\ 0 \\ 4 \end{bmatrix} \qquad y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 0 \\ 4 \end{bmatrix}$$

FIG. 27

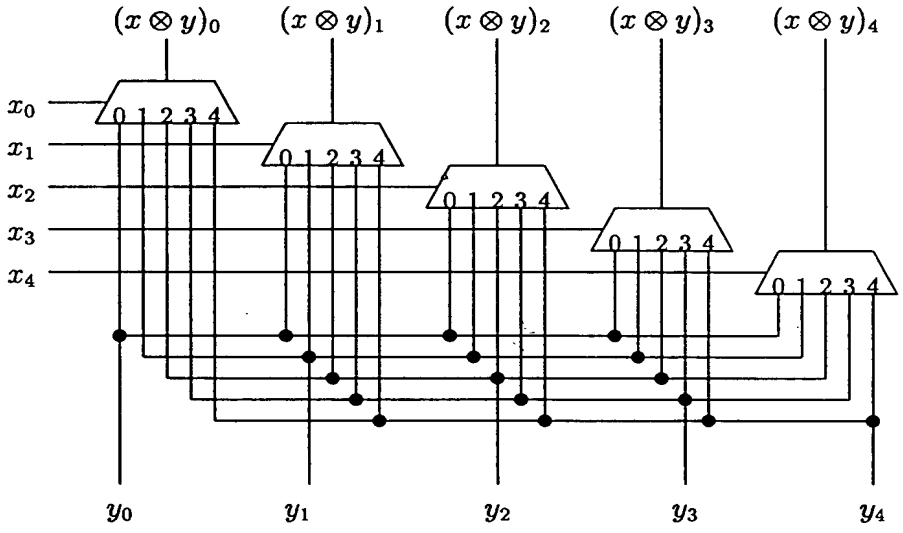
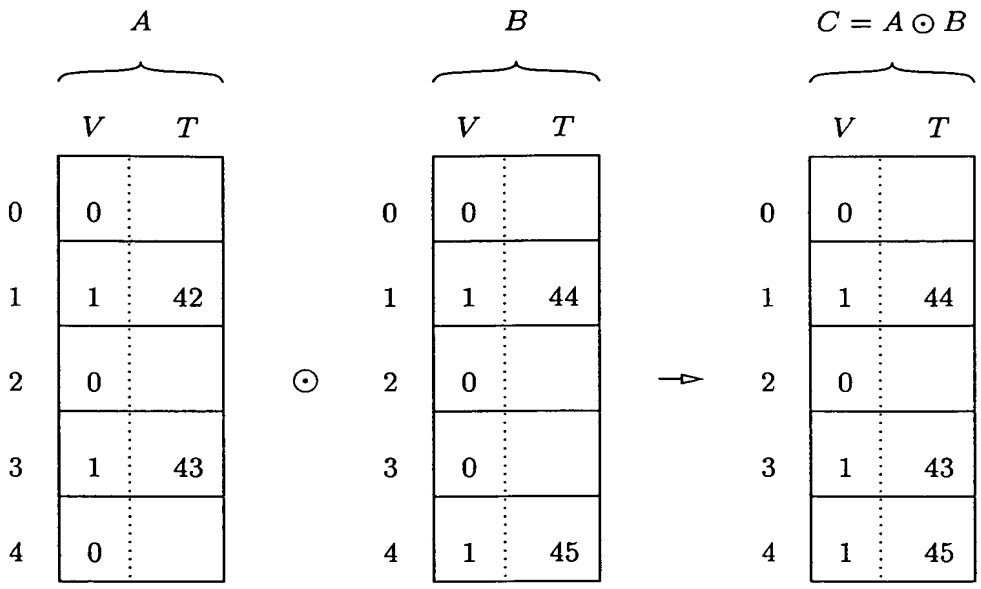

$(x \otimes y)_0$
$(x \otimes y)_1$
$(x \otimes y)_2$
$(x \otimes y)_3$
$(x \otimes y)_4$
x_0
x_1
x_2
x_3
x_4
0 1 2 3 4
y_0
y_1
y_2
y_3
y_4

FIG. 28



Logic for one row:

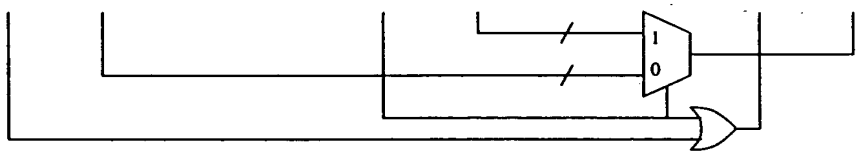


FIG. 29

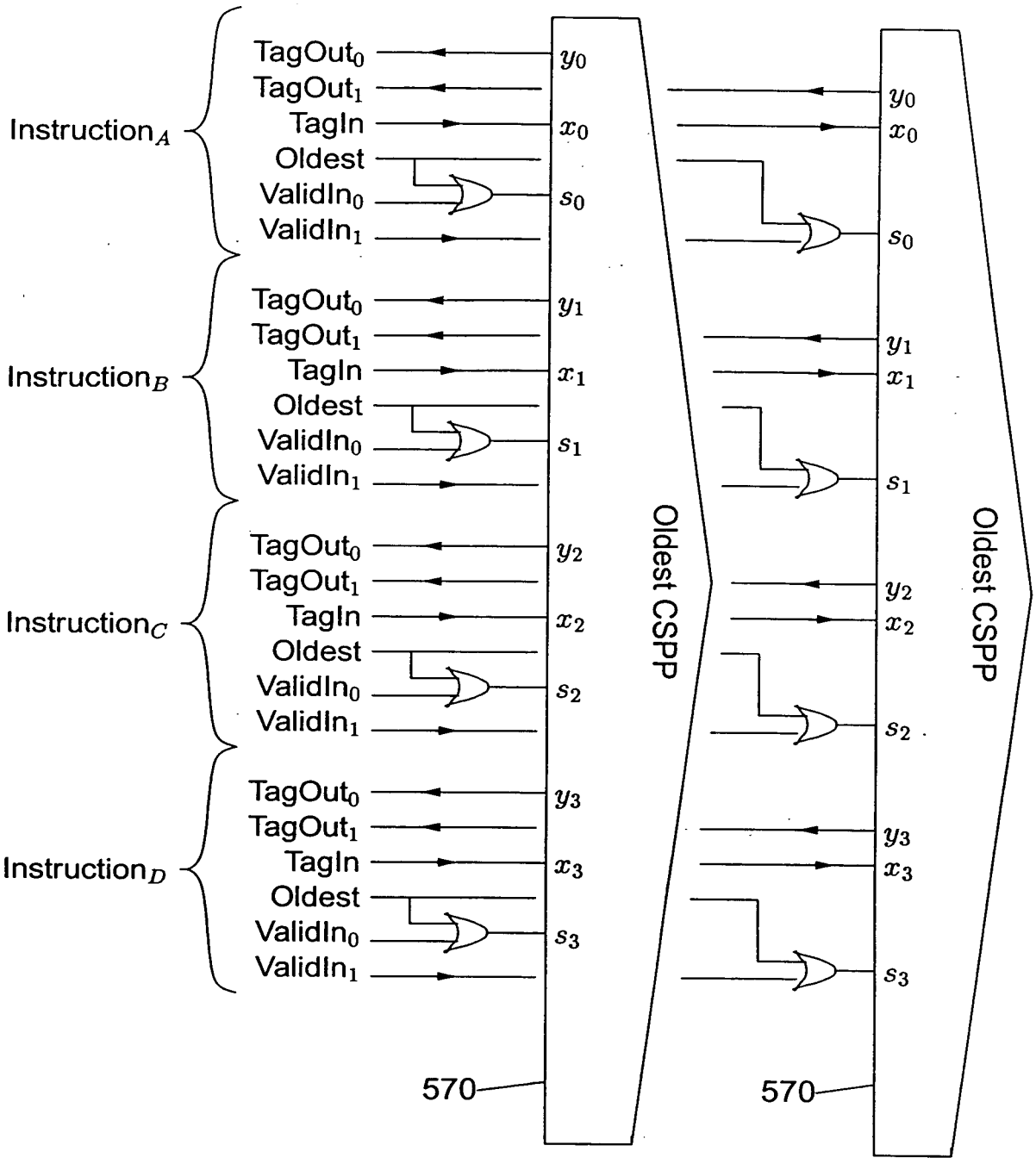

Instruction$_A$
Instruction$_B$
Instruction$_C$
Instruction$_D$
TagOut$_0$
TagOut$_1$
TagIn
Oldest
ValidIn$_0$
ValidIn$_1$
y_0
x_0
s_0
y_1
x_1
s_1
y_2
x_2
s_2
y_3
x_3
s_3
Oldest CSPP
570

FIG. 30

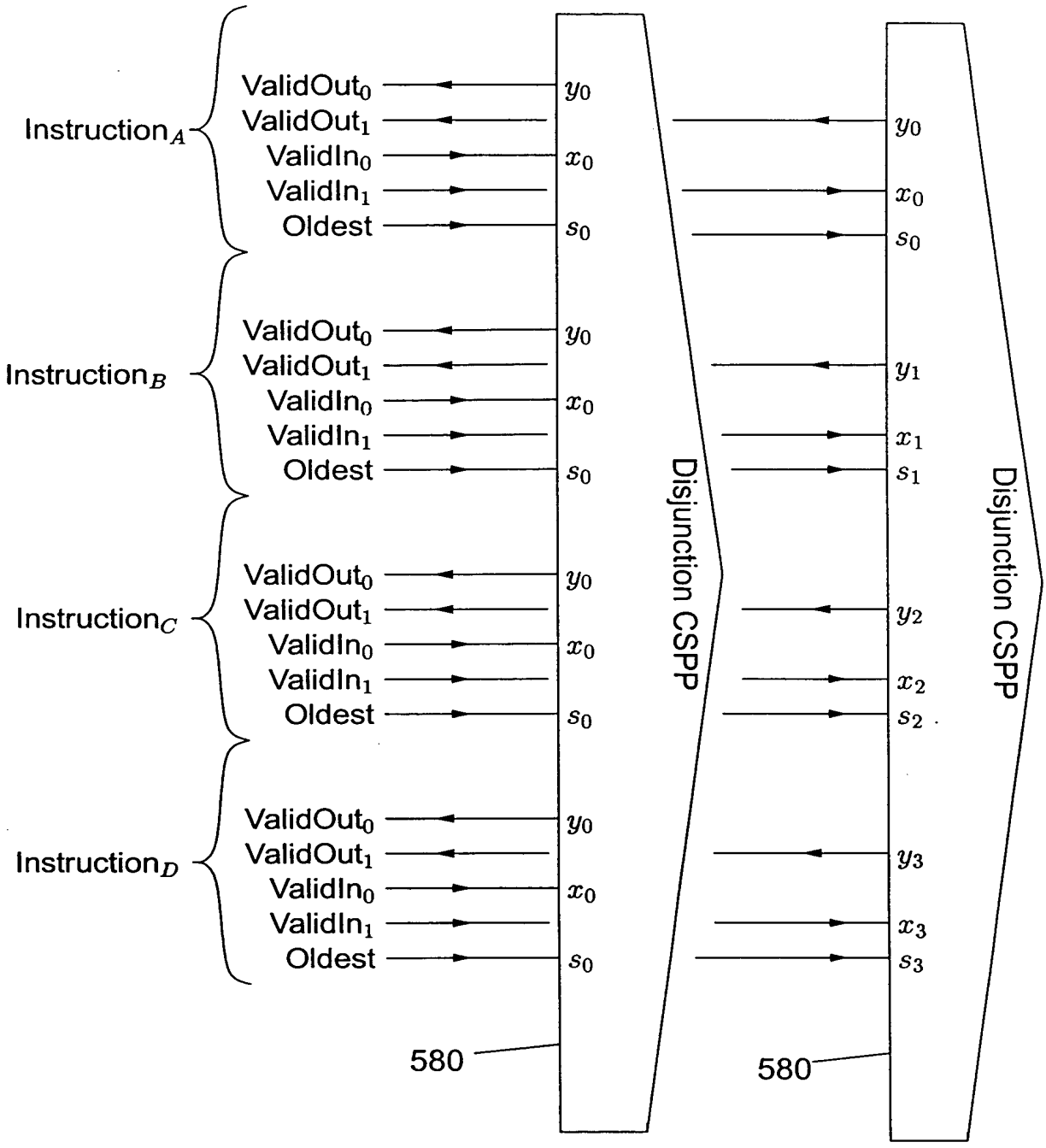

Instruction$_A$
Instruction$_B$
Instruction$_C$
Instruction$_D$
ValidOut$_0$
ValidOut$_1$
ValidIn$_0$
ValidIn$_1$
Oldest
y_0
x_0
s_0
Disjunction CSPP
y_1
x_1
s_1
y_2
x_2
s_2
y_3
x_3
s_3
580
580

FIG. 31

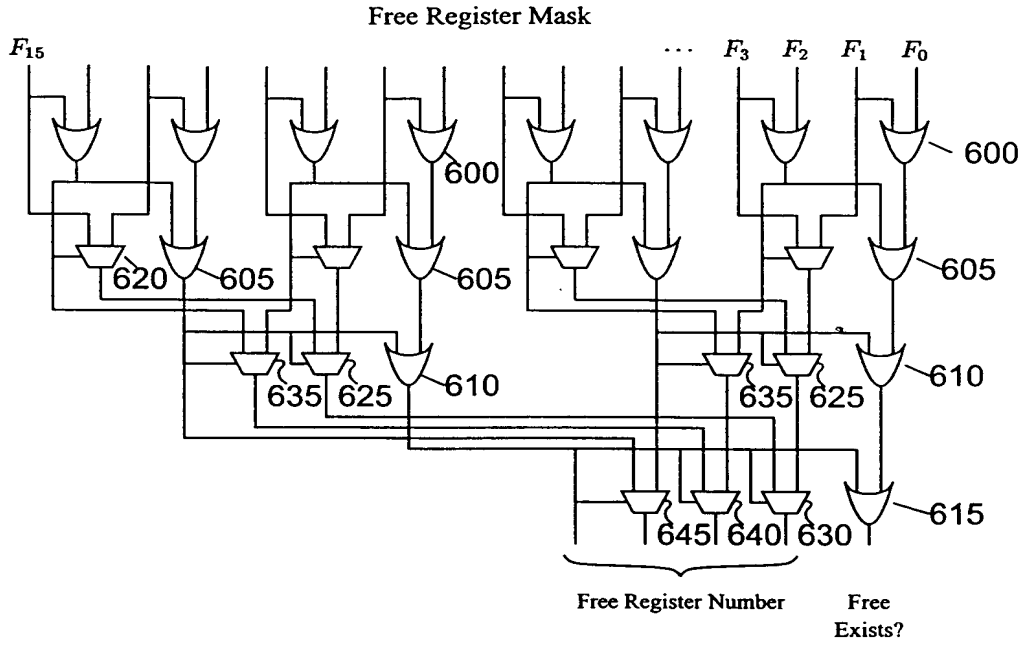

Free Register Mask
F_{15}
F_3
F_2
F_1
F_0
600
600
605
620
605
605
610
635
625
610
635
625
615
645
640
630
Free Register Number
Free Exists?

FIG. 32

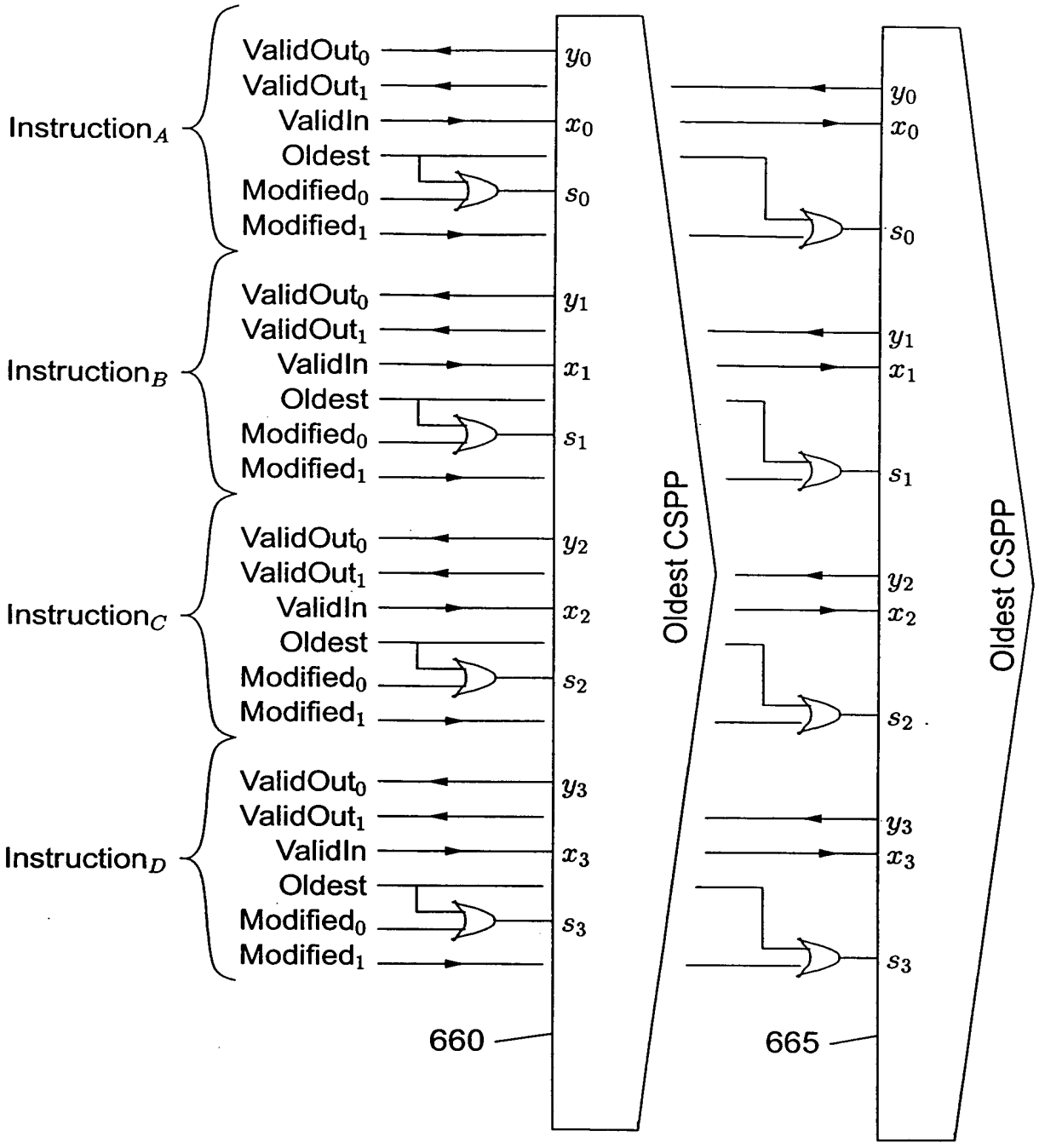

Instruction$_A$
Instruction$_B$
Instruction$_C$
Instruction$_D$
ValidOut$_0$
ValidOut$_1$
ValidIn
Oldest
Modified$_0$
Modified$_1$
y_0
x_0
s_0
y_1
x_1
s_1
y_2
x_2
s_2
y_3
x_3
s_3
Oldest CSPP
660
665

FIG. 33

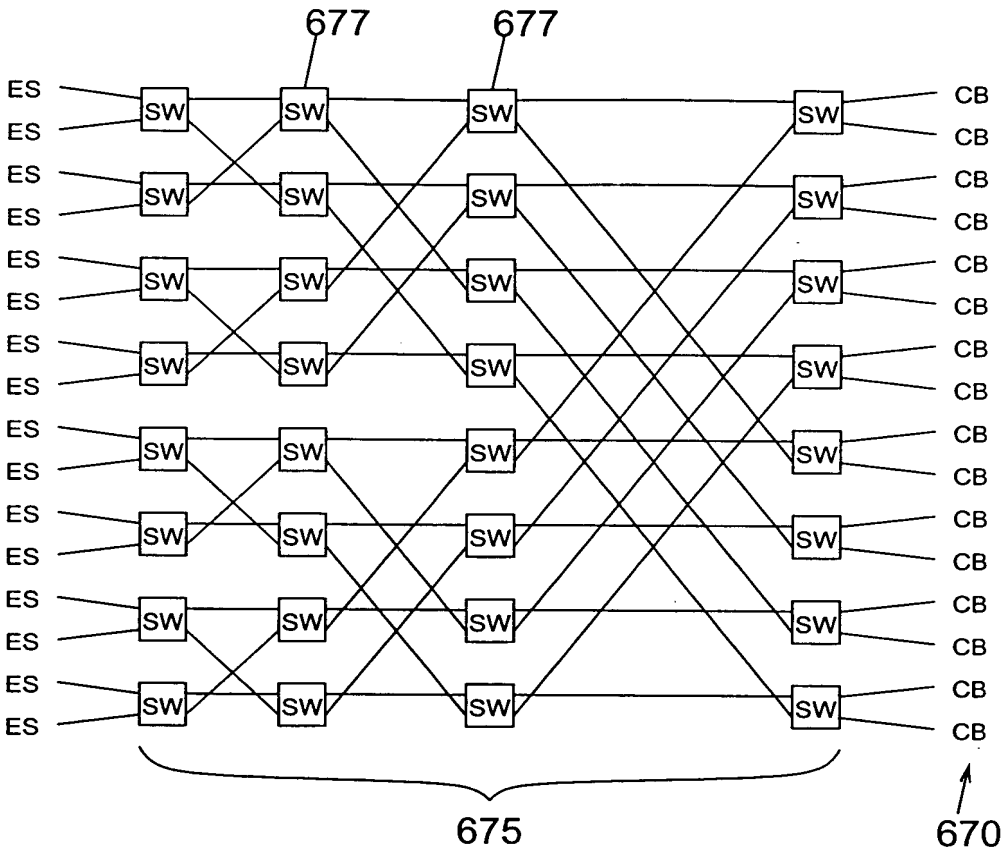


FIG. 34

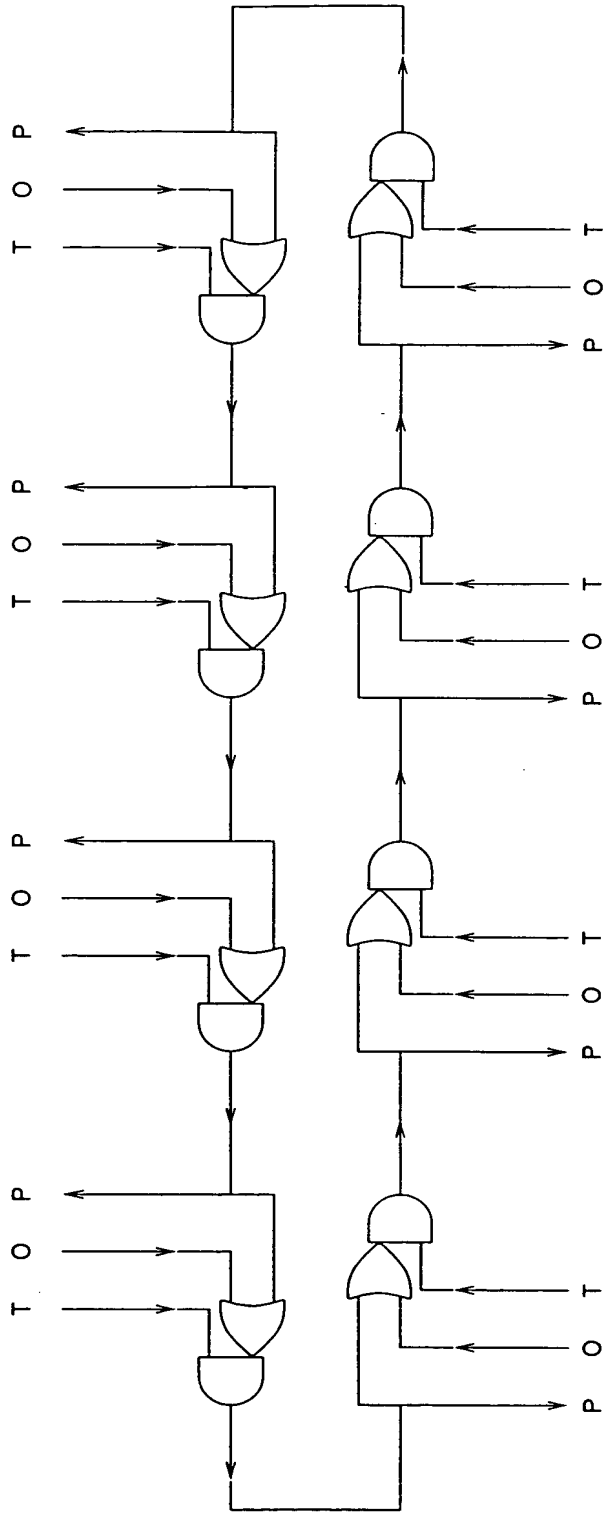


FIG. 35

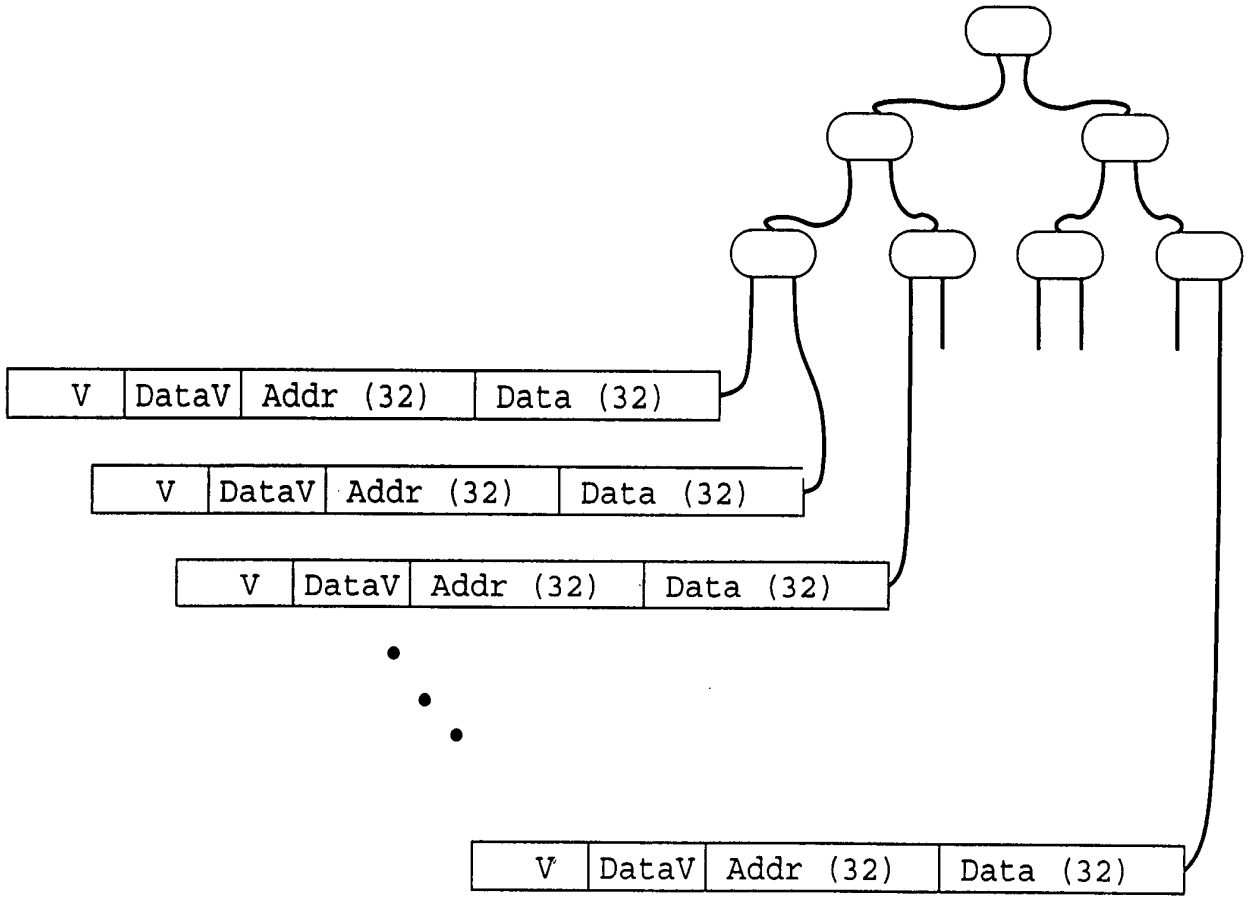
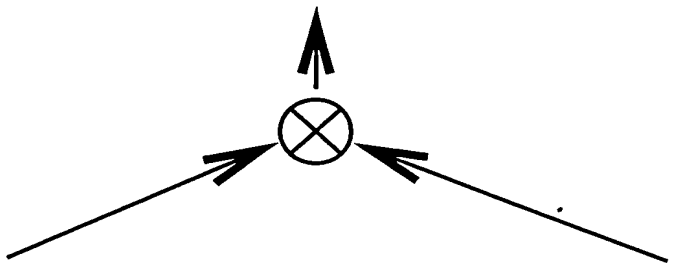

V
DataV
Addr (32)
Data (32)
V
DataV
Addr (32)
Data (32)
V
DataV
Addr (32)
Data (32)
V
DataV
Addr (32)
Data (32)

FIG. 36

V	Addr	DataV	Data
1	2	0	X
1	10	1	1
1	42	1	100
1	7	1	5



V	Addr	DataV	Data
1	2	0	X
1	10	1	1
1	42	1	100
0	X	X	X

V	Addr	DataV	Data
1	10	0	X
1	7	1	5
0	X	X	X
0	X	X	X

FIG. 37

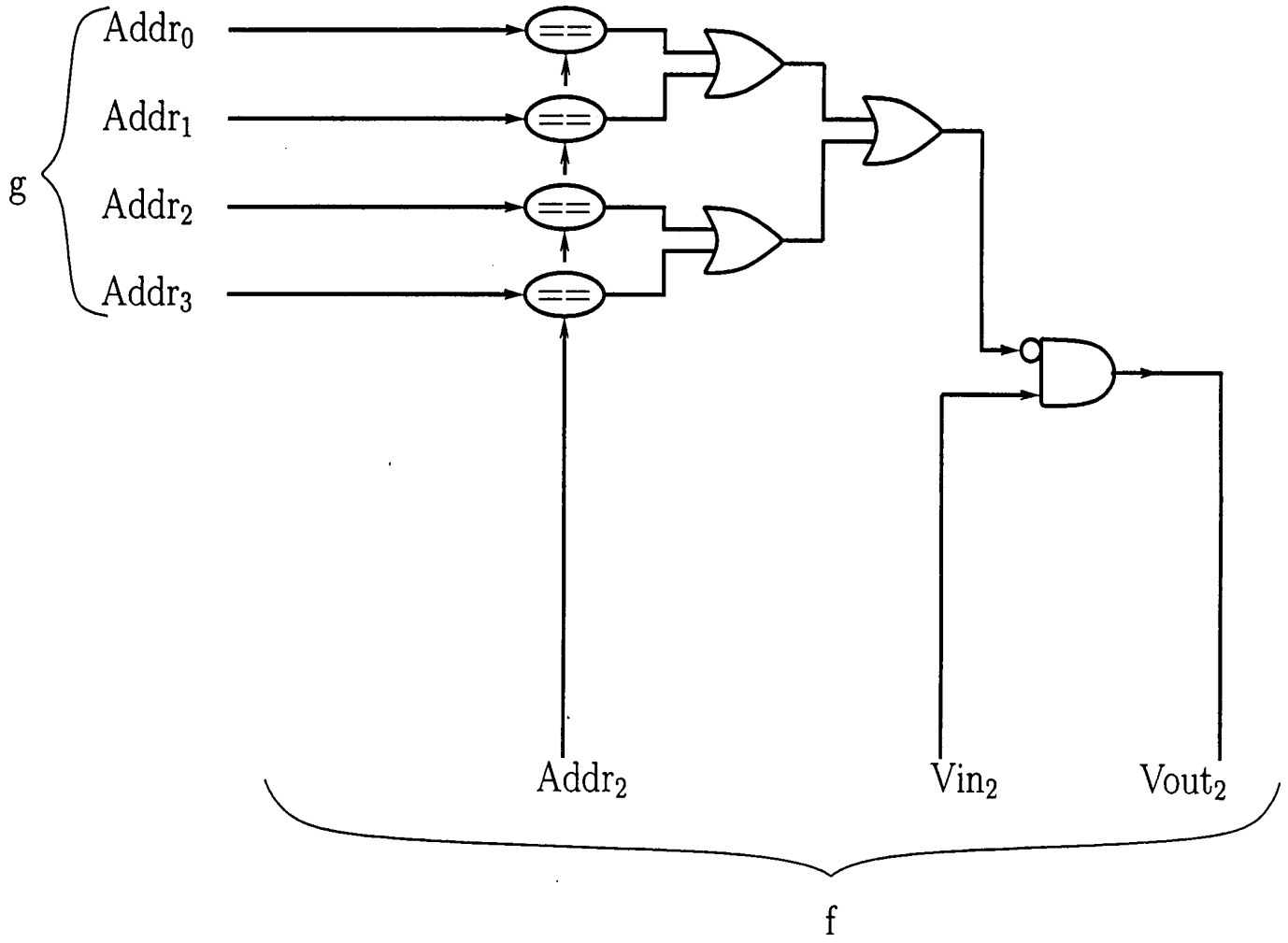


FIG. 38

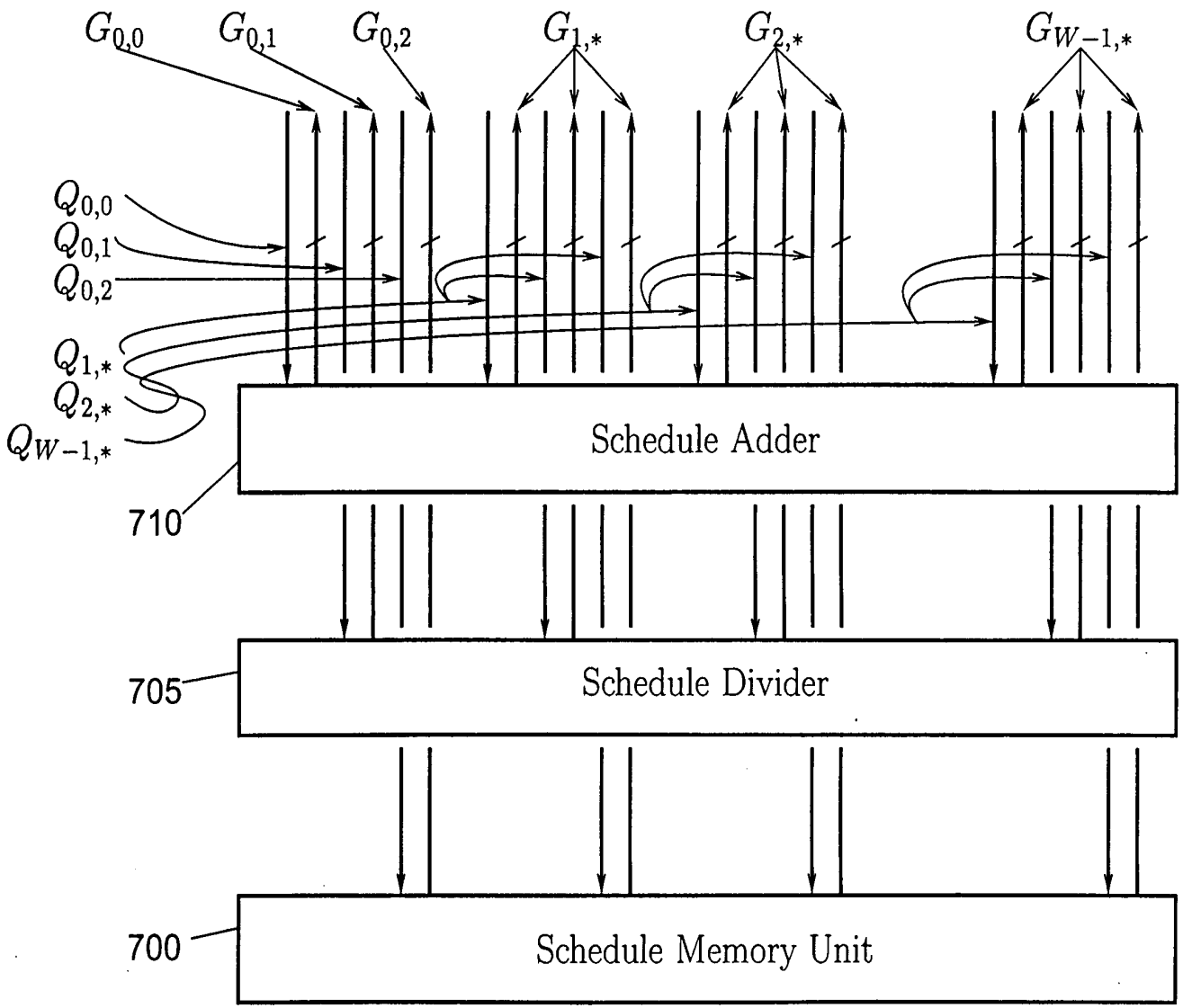

$G_{0,0}$
$G_{0,1}$
$G_{0,2}$
$G_{1,*}$
$G_{2,*}$
$G_{W-1,*}$
$Q_{0,0}$
$Q_{0,1}$
$Q_{0,2}$
$Q_{1,*}$
$Q_{2,*}$
$Q_{W-1,*}$
Schedule Adder
710
Schedule Divider
705
Schedule Memory Unit
700

FIG. 39

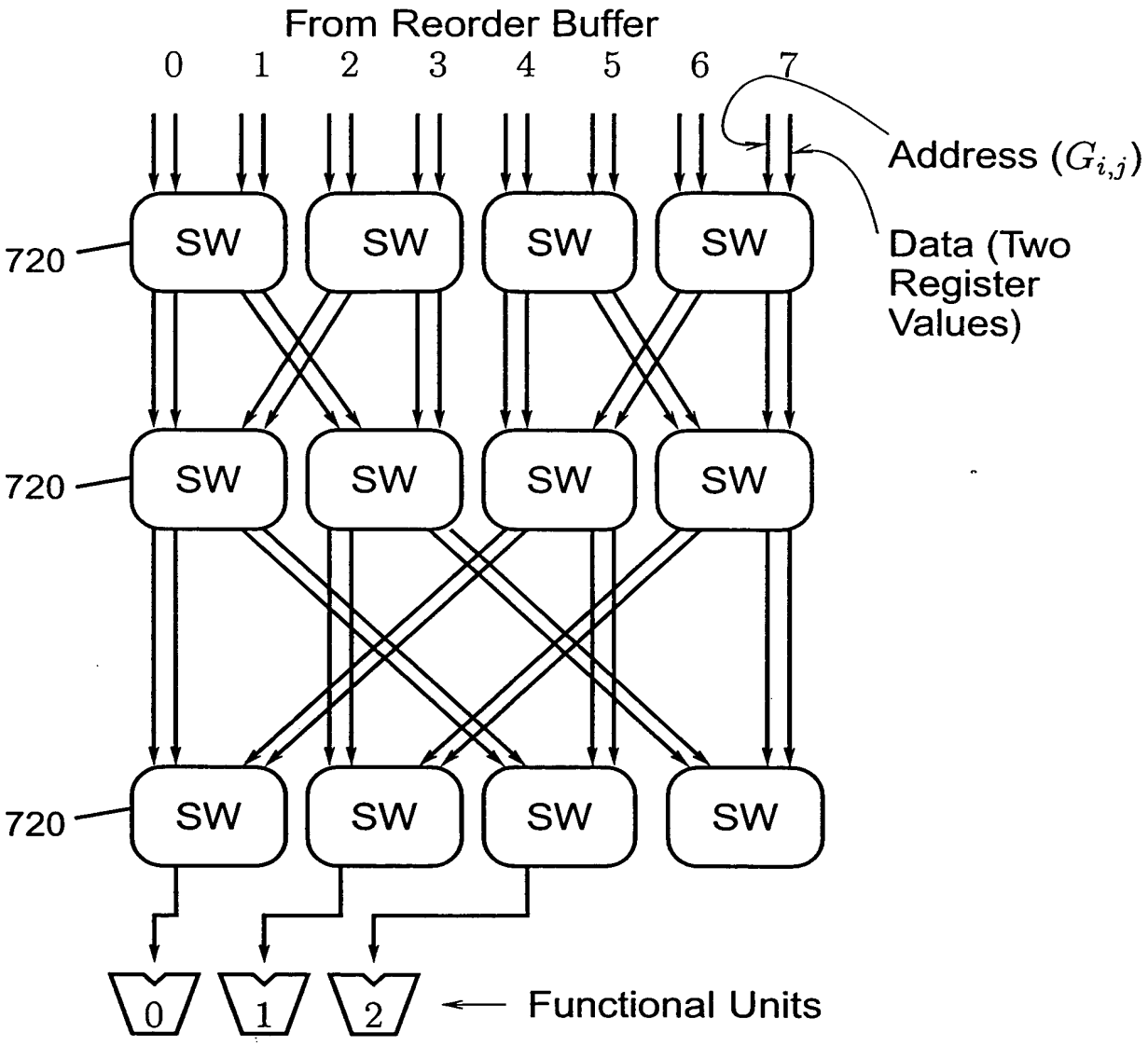

From Reorder Buffer
0
1
2
3
4
5
6
7
Address ($G_{i,j}$)
Data (Two Register Values)
720
SW
720
SW
720
SW
0
1
2
Functional Units

FIG. 40

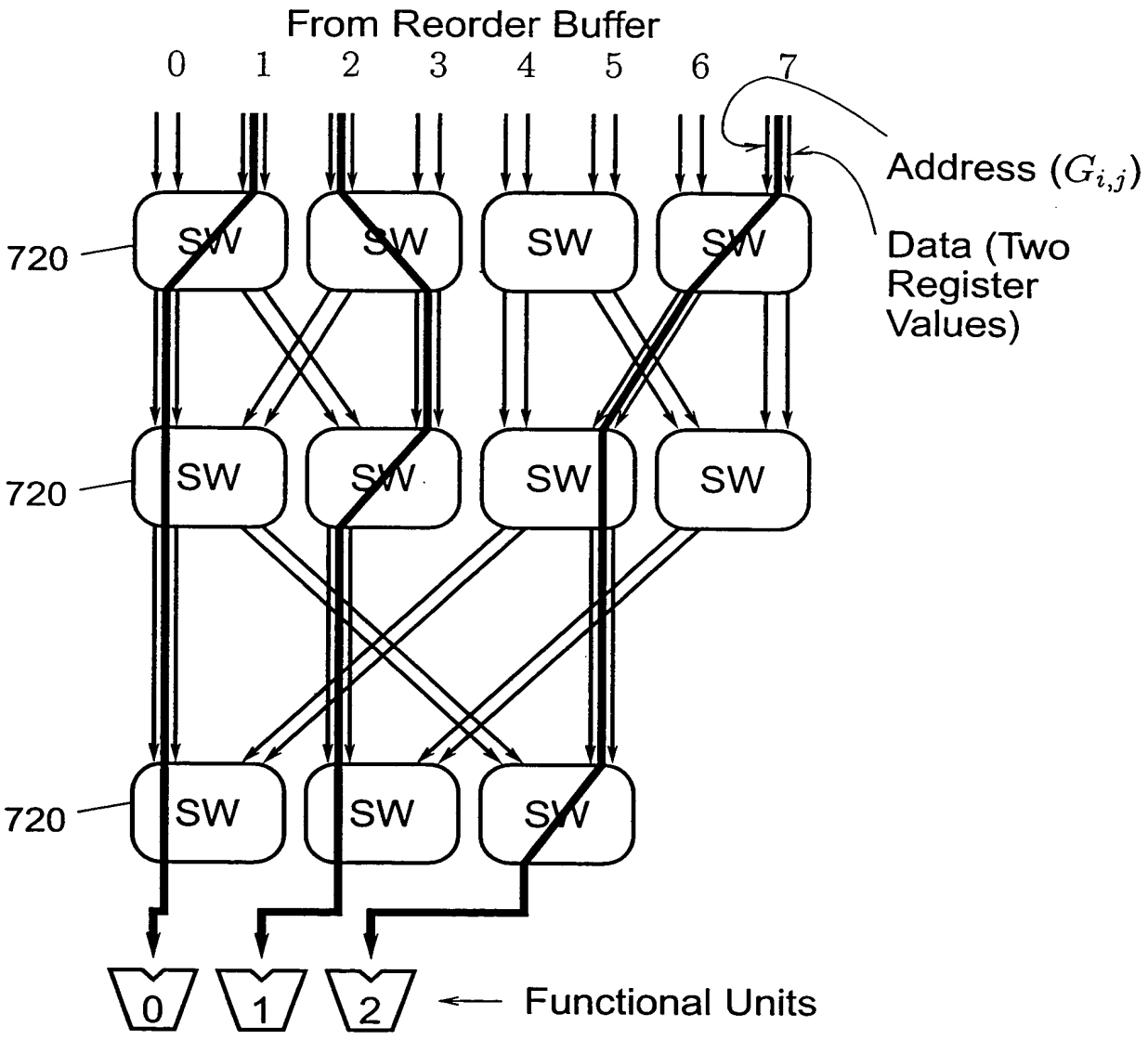

From Reorder Buffer
0
1
2
3
4
5
6
7
Address ($G_{i,j}$)
Data (Two Register Values)
720
720
720
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
SW
0
1
2
Functional Units

FIG. 41

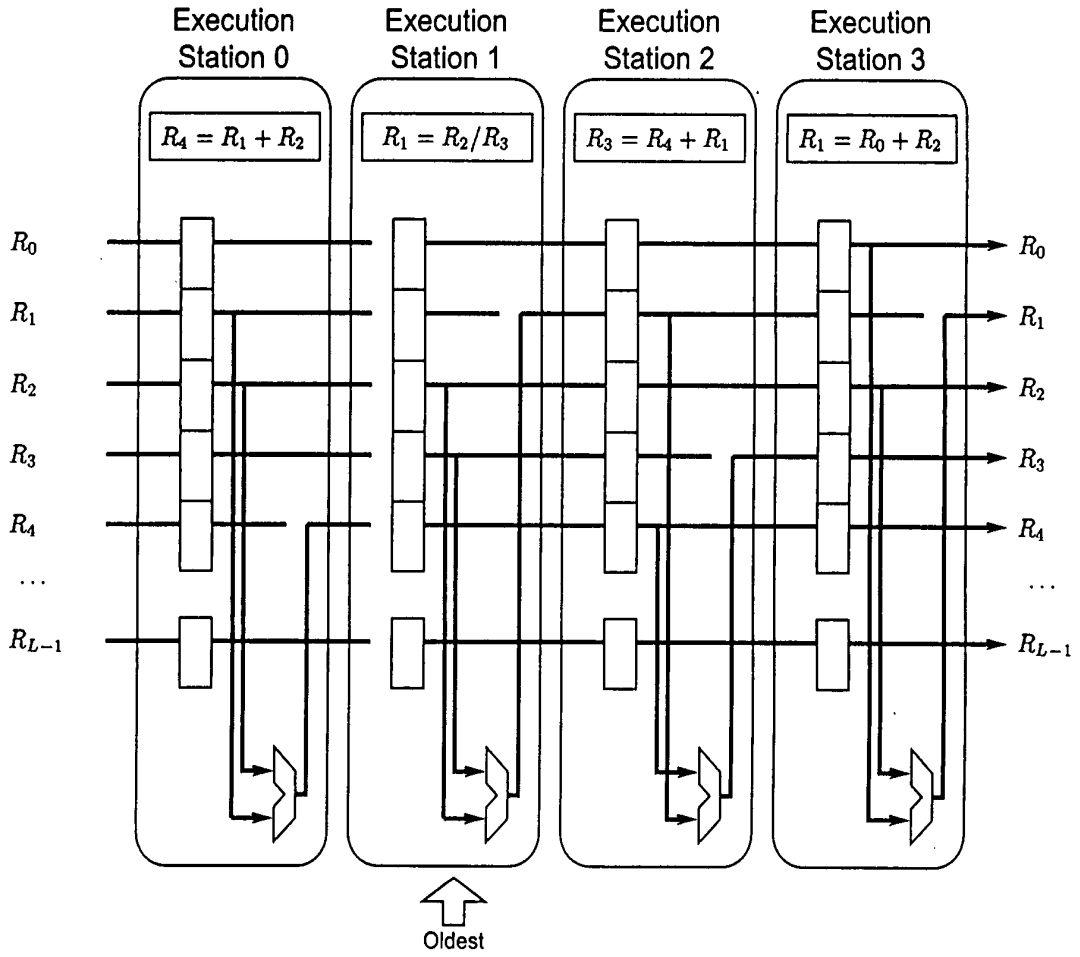

Execution Station 0
Execution Station 1
Execution Station 2
Execution Station 3
$R_4 = R_1 + R_2$
$R_1 = R_2/R_3$
$R_3 = R_4 + R_1$
$R_1 = R_0 + R_2$
R_0
R_1
R_2
R_3
R_4
...
R_{L-1}
Oldest

FIG. 42

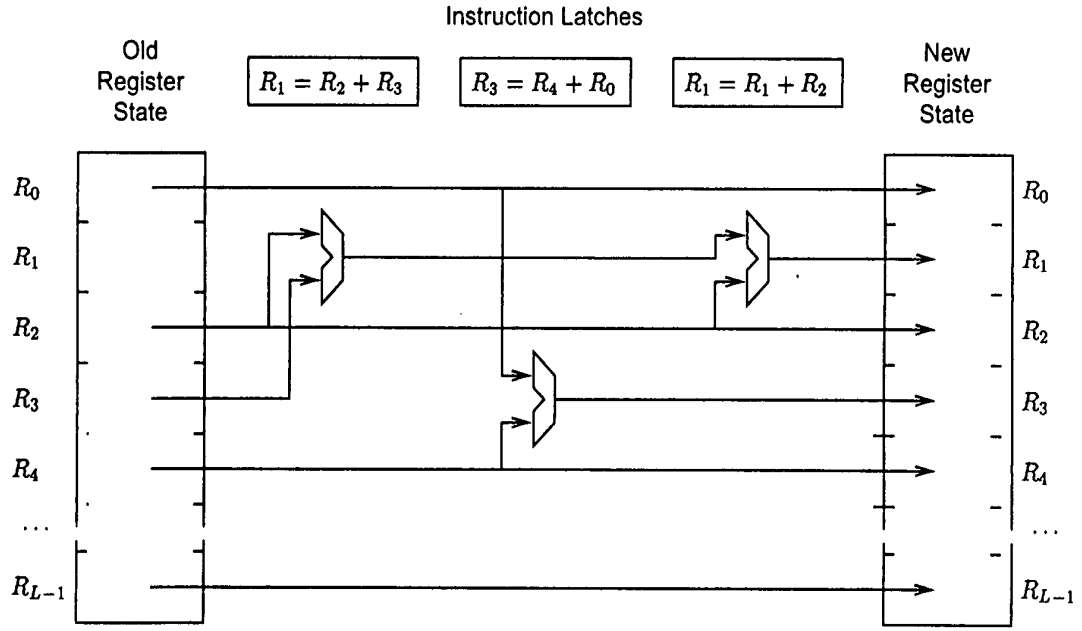

Instruction Latches
Old Register State
$R_1 = R_2 + R_3$
$R_3 = R_4 + R_0$
$R_1 = R_1 + R_2$
New Register State
R_0
R_1
R_2
R_3
R_4
...
R_{L-1}
R_0
R_1
R_2
R_3
R_4
...
R_{L-1}

FIG. 43

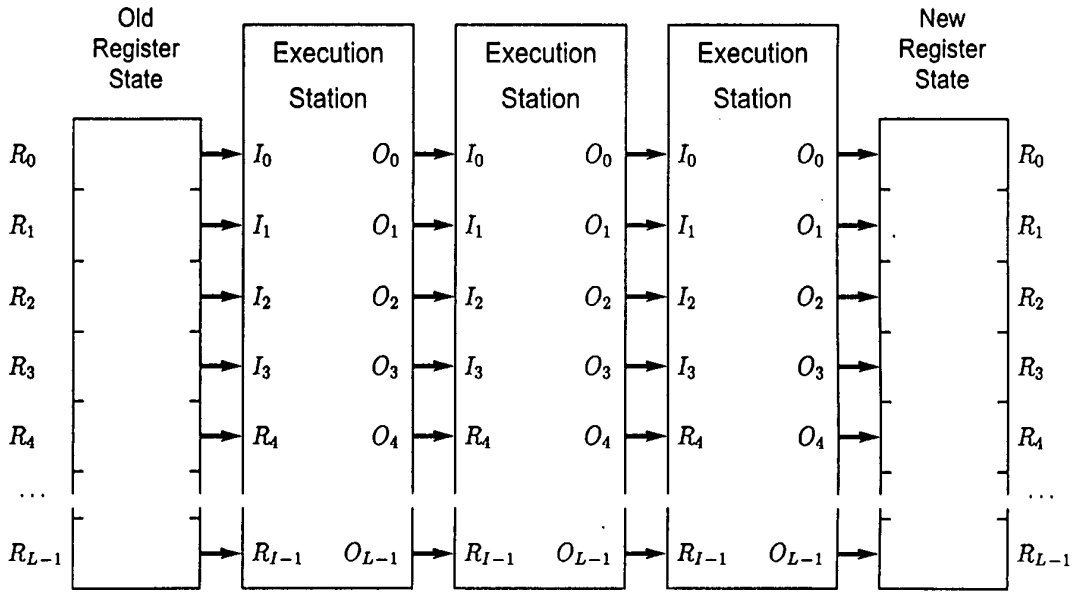

Old
Register
State
Execution
Station
Execution
Station
Execution
Station
New
Register
State
R_0
R_1
R_2
R_3
R_4
...
R_{L-1}
I_0
I_1
I_2
I_3
R_4
R_{I-1}
O_0
O_1
O_2
O_3
O_4
O_{L-1}

FIG. 44

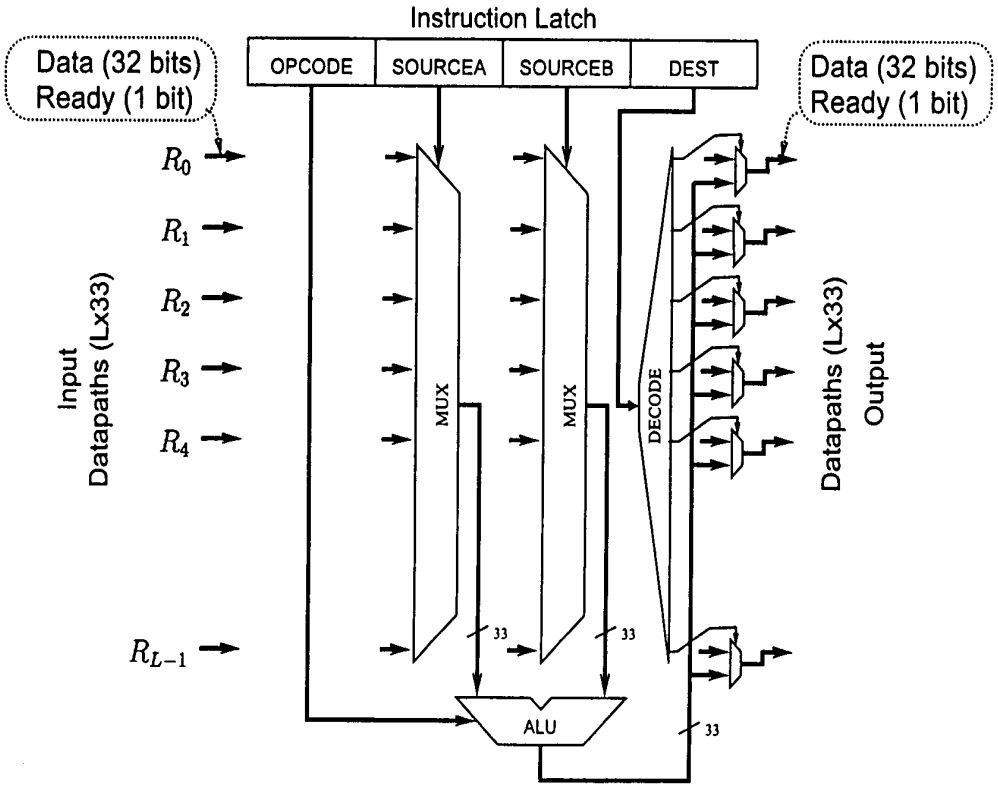

Instruction Latch
OPCODE
SOURCEA
SOURCEB
DEST
Data (32 bits)
Ready (1 bit)
Data (32 bits)
Ready (1 bit)
R_0
R_1
R_2
R_3
R_4
R_{L-1}
Input
Datapaths (Lx33)
MUX
MUX
DECODE
33
33
33
ALU
Datapaths (Lx33)
Output

FIG. 45

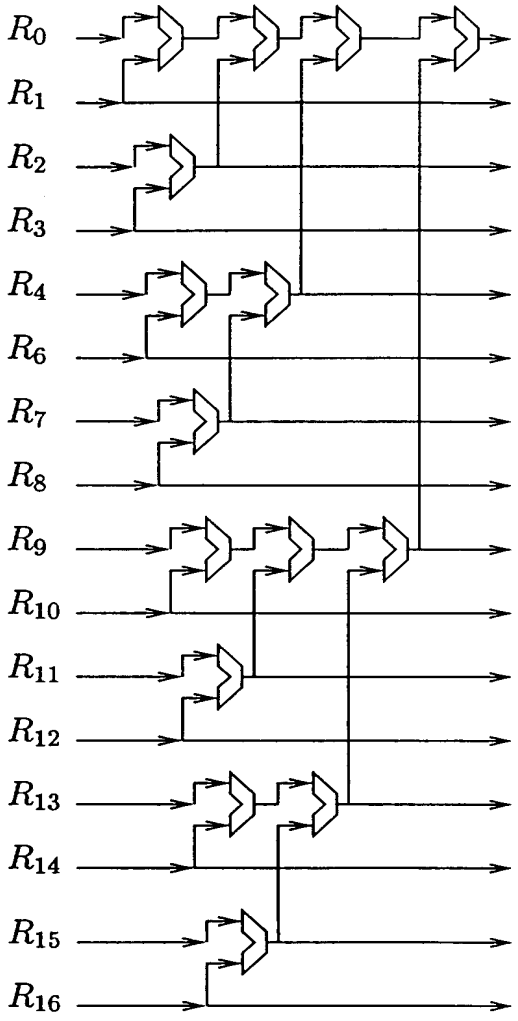


FIG. 46

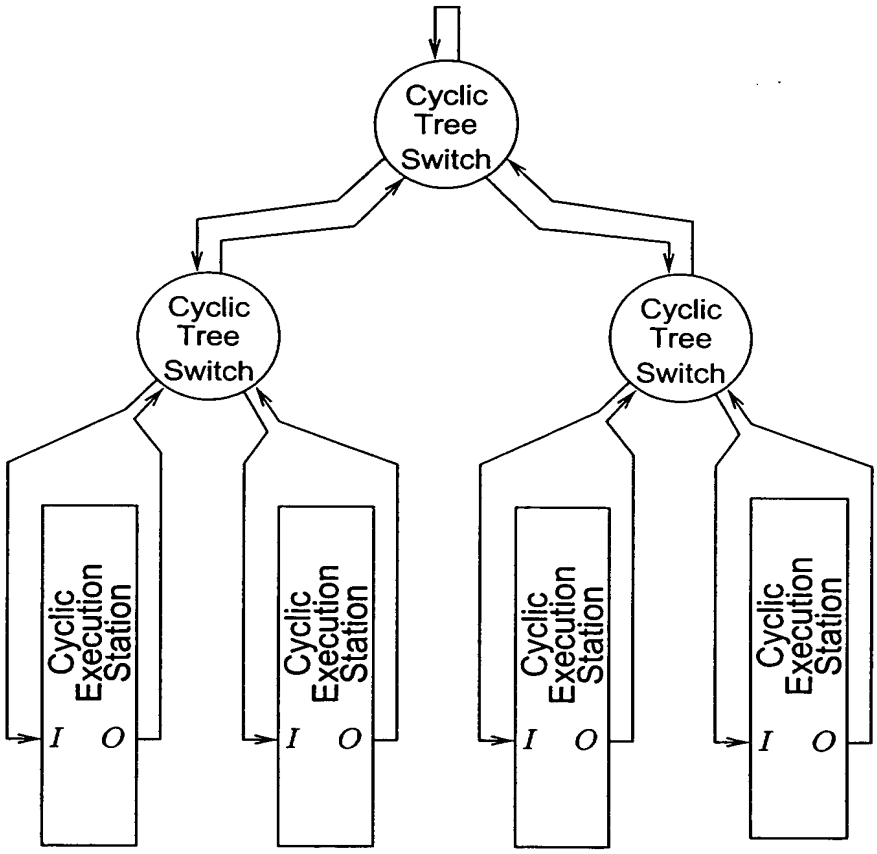

Cyclic
Tree
Switch
Cyclic
Tree
Switch
Cyclic
Tree
Switch
Cyclic
Execution
Station
I O
Cyclic
Execution
Station
I O
Cyclic
Execution
Station
I O
Cyclic
Execution
Station
I O

FIG. 47

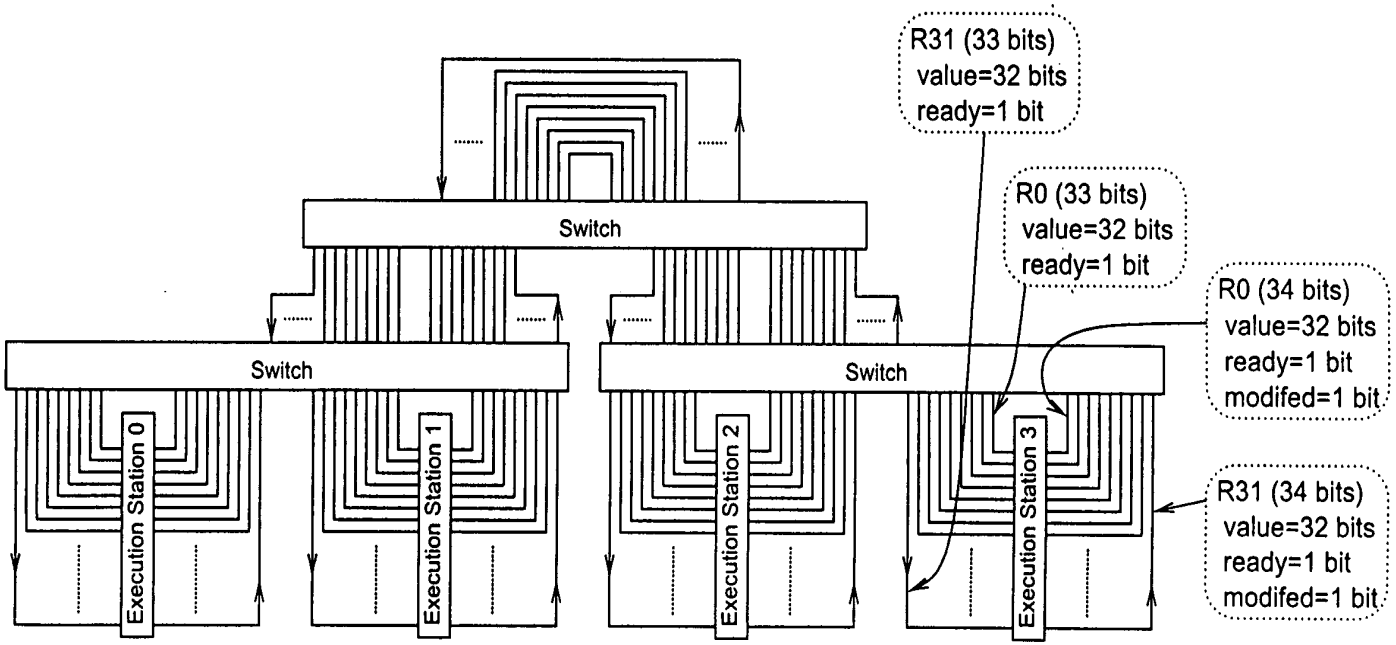

R31 (33 bits)
value=32 bits
ready=1 bit
R0 (33 bits)
value=32 bits
ready=1 bit
R0 (34 bits)
value=32 bits
ready=1 bit
modifed=1 bit
R31 (34 bits)
value=32 bits
ready=1 bit
modifed=1 bit
Switch
Switch
Switch
Execution Station 0
Execution Station 1
Execution Station 2
Execution Station 3

FIG. 48

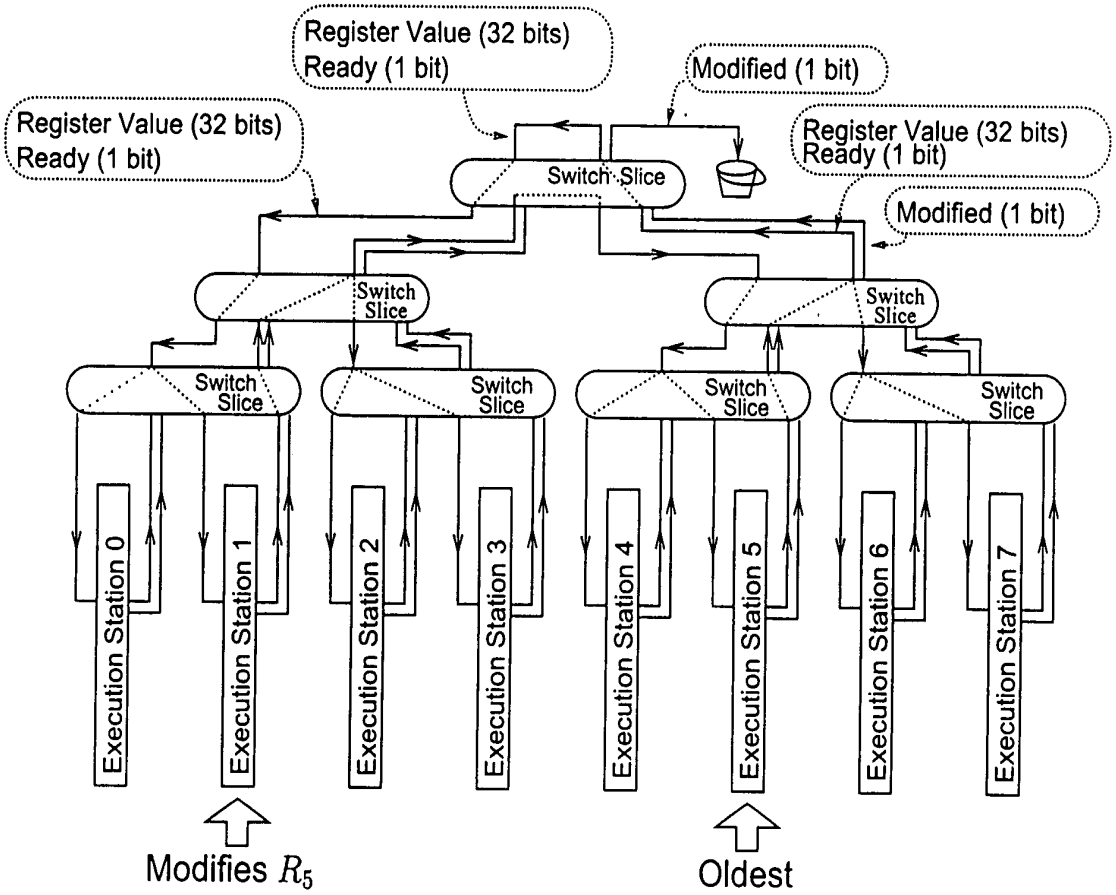

Register Value (32 bits)
Ready (1 bit)
Modified (1 bit)
Register Value (32 bits)
Ready (1 bit)
Register Value (32 bits)
Ready (1 bit)
Modified (1 bit)
Switch Slice
Switch Slice
Switch Slice
Switch Slice
Switch Slice
Switch Slice
Switch Slice
Execution Station 0
Execution Station 1
Execution Station 2
Execution Station 3
Execution Station 4
Execution Station 5
Execution Station 6
Execution Station 7
Modifies R_5
Oldest

FIG. 49

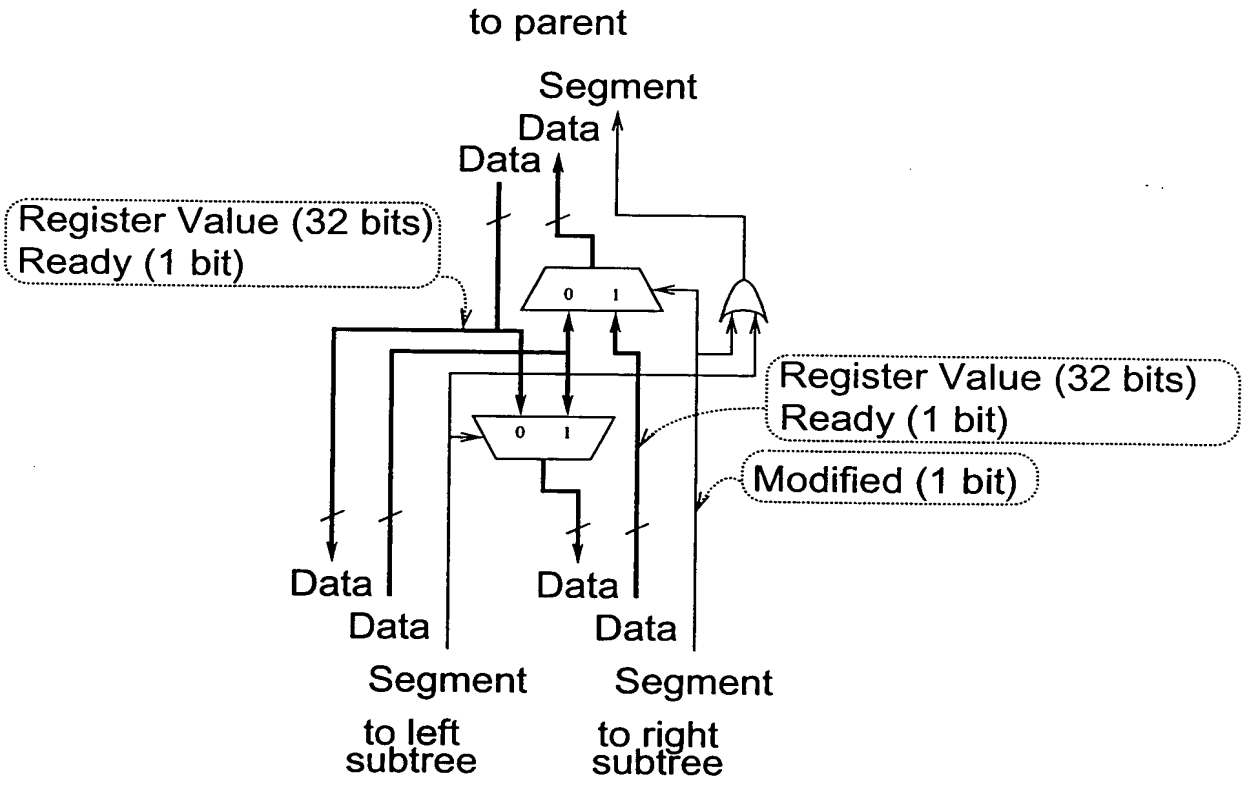

to parent
Segment
Data
Data
Register Value (32 bits)
Ready (1 bit)
0 1
Register Value (32 bits)
Ready (1 bit)
0 1
Modified (1 bit)
Data
Data
Data
Data
Segment
to left
subtree
Segment
to right
subtree

FIG. 50

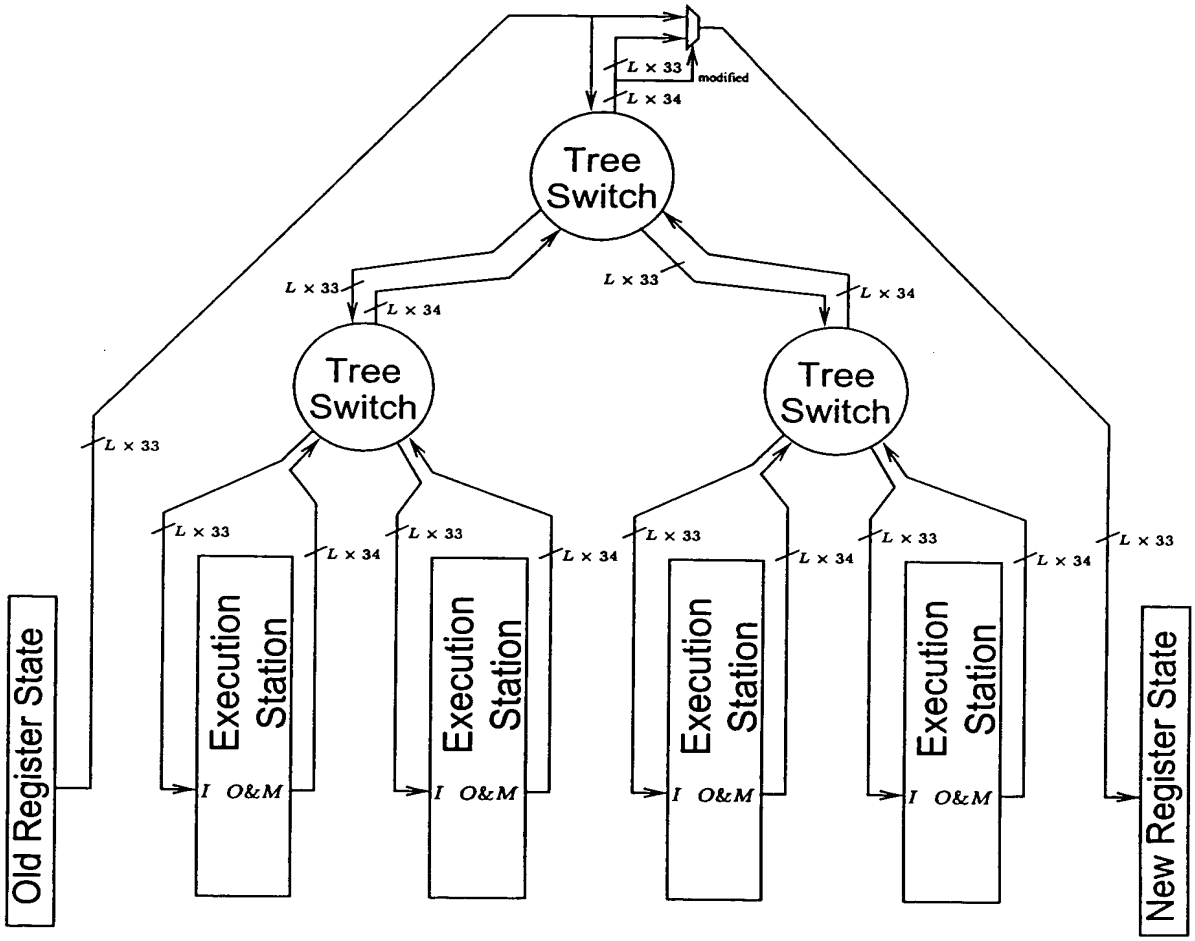

L × 33
modified
L × 34
Tree Switch
L × 33
L × 34
L × 33
L × 34
Tree Switch
Tree Switch
L × 33
L × 33
L × 34
L × 33
L × 34
L × 33
L × 34
L × 33
L × 34
Execution Station
I O&M
Execution Station
I O&M
Execution Station
I O&M
Execution Station
I O&M
Old Register State
New Register State

FIG. 51

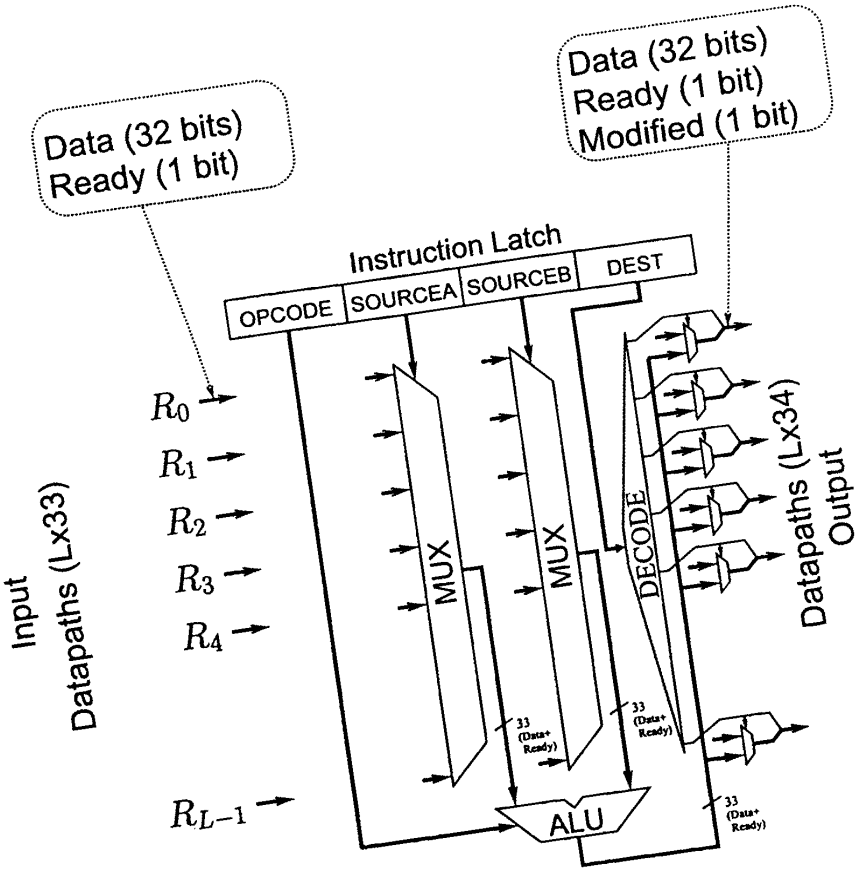

Data (32 bits)
Ready (1 bit)
Data (32 bits)
Ready (1 bit)
Modified (1 bit)
Instruction Latch
OPCODE
SOURCEA
SOURCEB
DEST
R_0
R_1
R_2
R_3
R_4
R_{L-1}
Input
Datapaths (Lx33)
MUX
MUX
DECODE
ALU
33
(Data+
Ready)
33
(Data+
Ready)
33
(Data+
Ready)
Datapaths (Lx34)
Output

FIG. 52

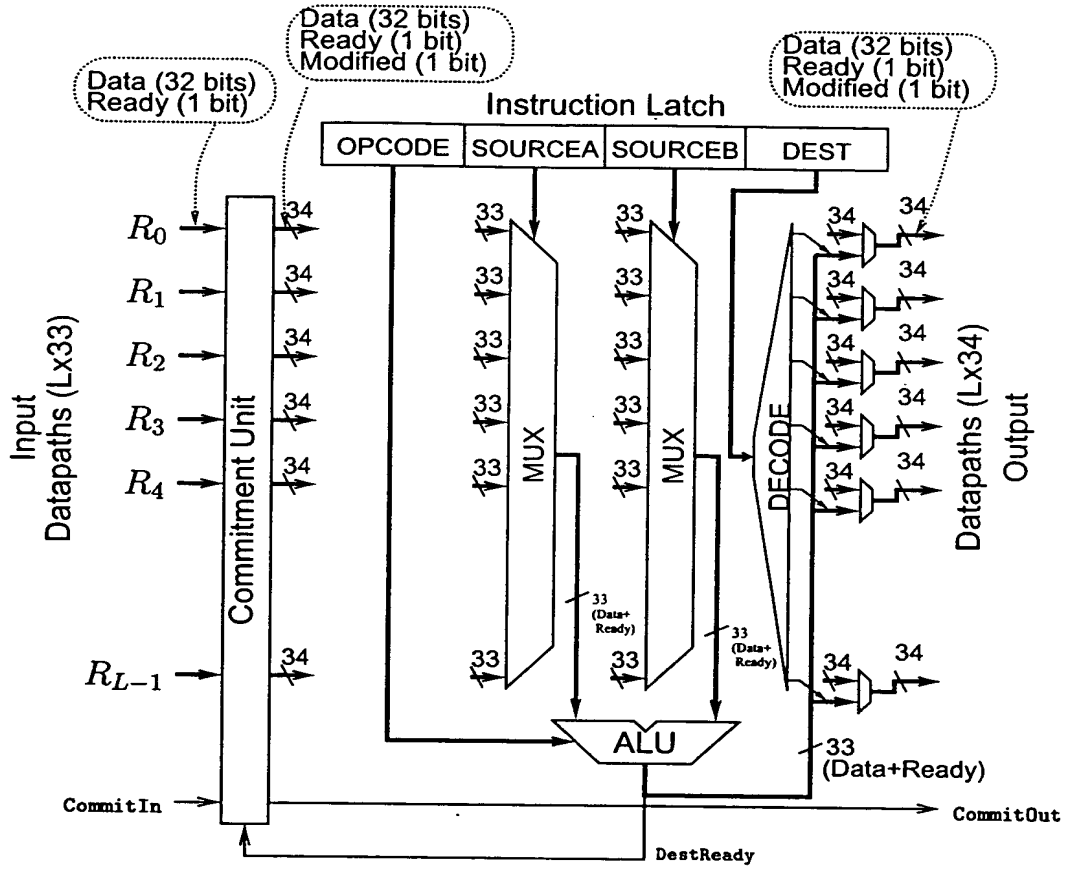

Data (32 bits)
Ready (1 bit)
Modified (1 bit)
Data (32 bits)
Ready (1 bit)
Data (32 bits)
Ready (1 bit)
Modified (1 bit)
Instruction Latch
OPCODE
SOURCEA
SOURCEB
DEST
Input
Datapaths (Lx33)
R_0
R_1
R_2
R_3
R_4
R_{L-1}
Commitment Unit
MUX
MUX
DECODE
Datapaths (Lx34)
Output
ALU
33
(Data+Ready)
CommitIn
CommitOut
DestReady

FIG. 53

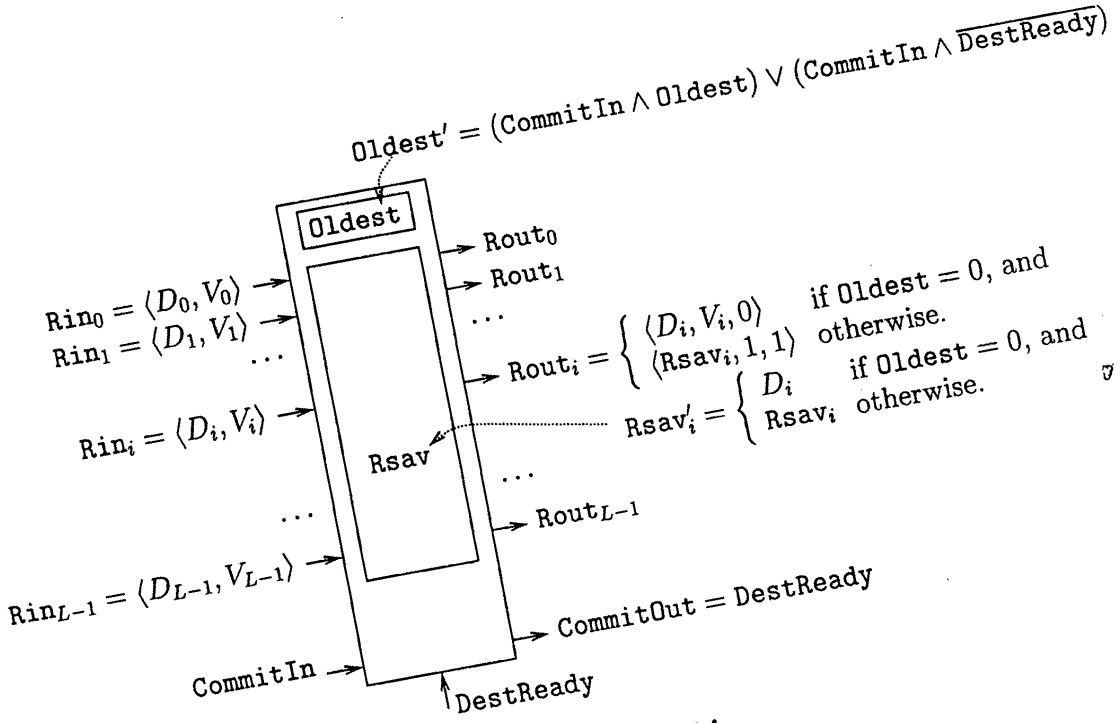

Oldest′ = (CommitIn ∧ Oldest) ∨ (CommitIn ∧ DestReady)
Oldest
Rout0
Rout1
...
Rin0 = ⟨D0, V0⟩
Rin1 = ⟨D1, V1⟩
...
Routi = ⟨Di, Vi, 0⟩ if Oldest = 0, and
⟨Rsavi, 1, 1⟩ otherwise.
Rini = ⟨Di, Vi⟩
Rsav′i = Di if Oldest = 0, and
Rsavi otherwise.
Rsav
...
...
RoutL−1
RinL−1 = ⟨DL−1, VL−1⟩
CommitOut = DestReady
CommitIn
DestReady

FIG. 54

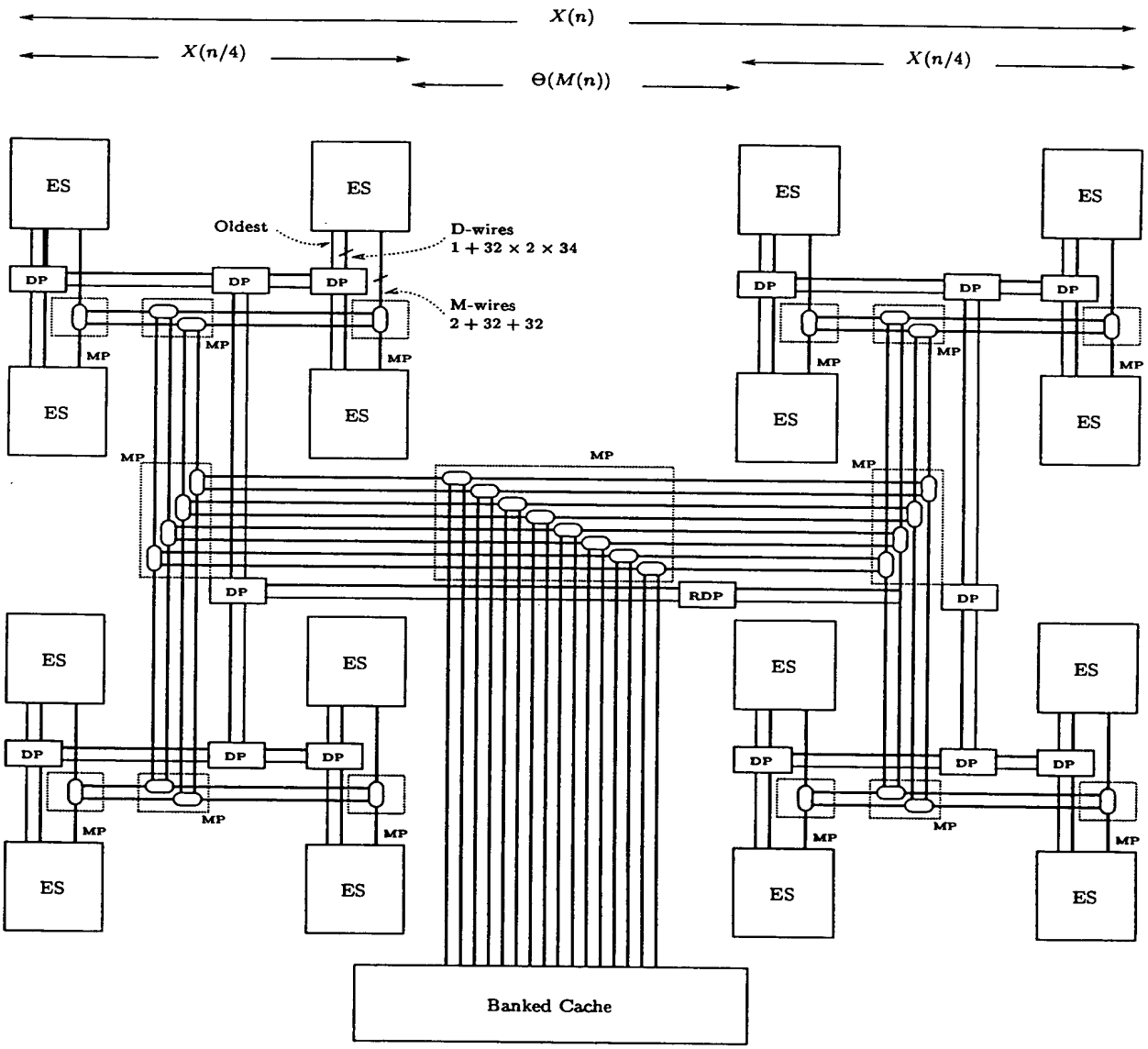

X(n)
X(n/4)
Θ(M(n))
X(n/4)
Oldest
D-wires
1 + 32 × 2 × 34
M-wires
2 + 32 + 32
ES
DP
MP
RDP
Banked Cache

FIG. 55

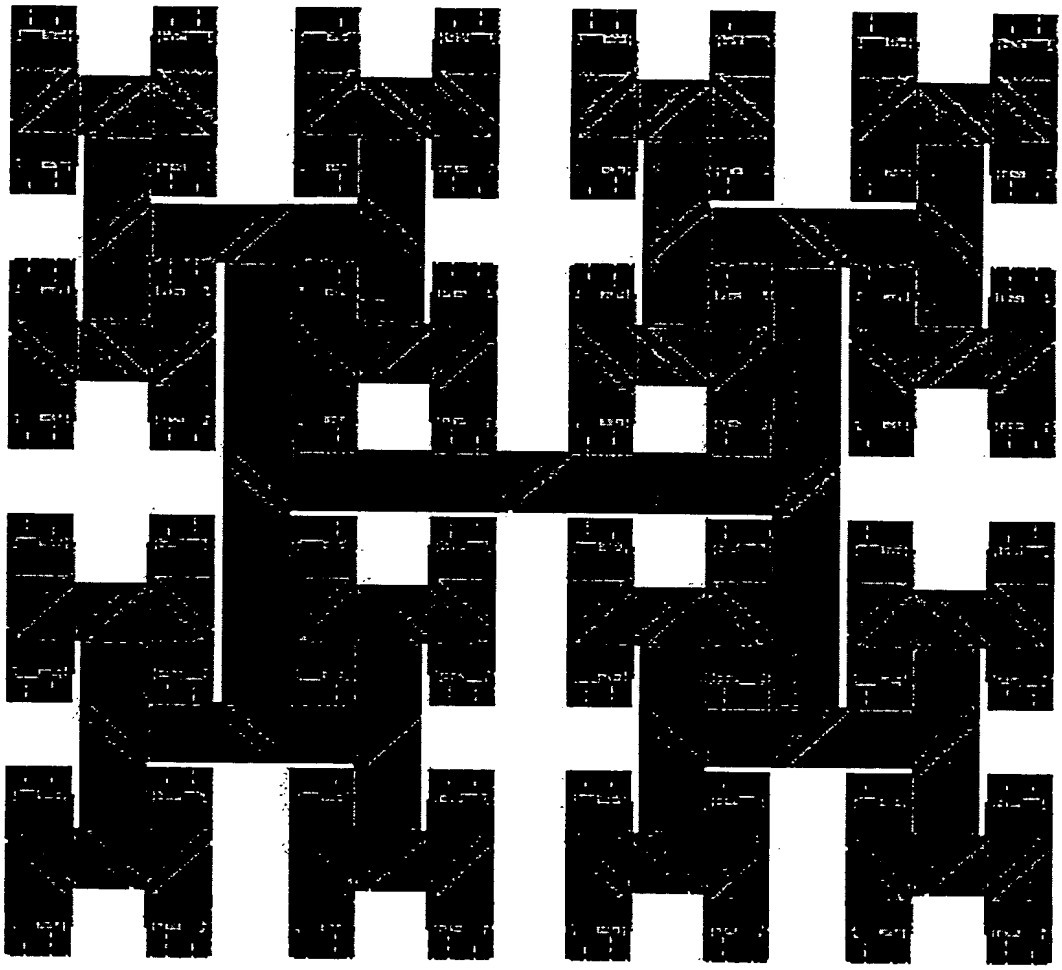


FIG. 56VLSI layout.

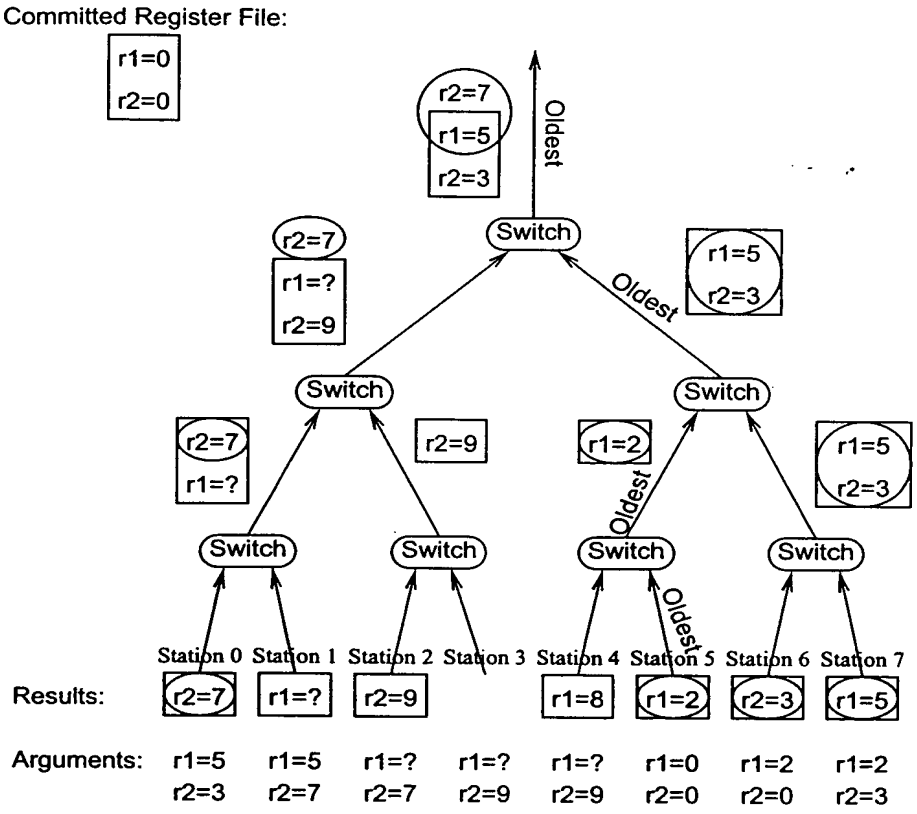

Committed Register File:
r1=0
r2=0
r2=7
r1=5
r2=3
Oldest
Switch
r2=7
r1=?
r2=9
r1=5
r2=3
Oldest
Switch
Switch
r2=7
r1=?
r2=9
r1=2
r1=5
r2=3
Oldest
Switch
Switch
Switch
Switch
Oldest
Station 0
Station 1
Station 2
Station 3
Station 4
Station 5
Station 6
Station 7
Results:
r2=7
r1=?
r2=9
r1=8
r1=2
r2=3
r1=5
Arguments:
r1=5 r2=3
r1=5 r2=7
r1=? r2=7
r1=? r2=9
r1=? r2=9
r1=0 r2=0
r1=2 r2=0
r1=2 r2=3

FIG. 57

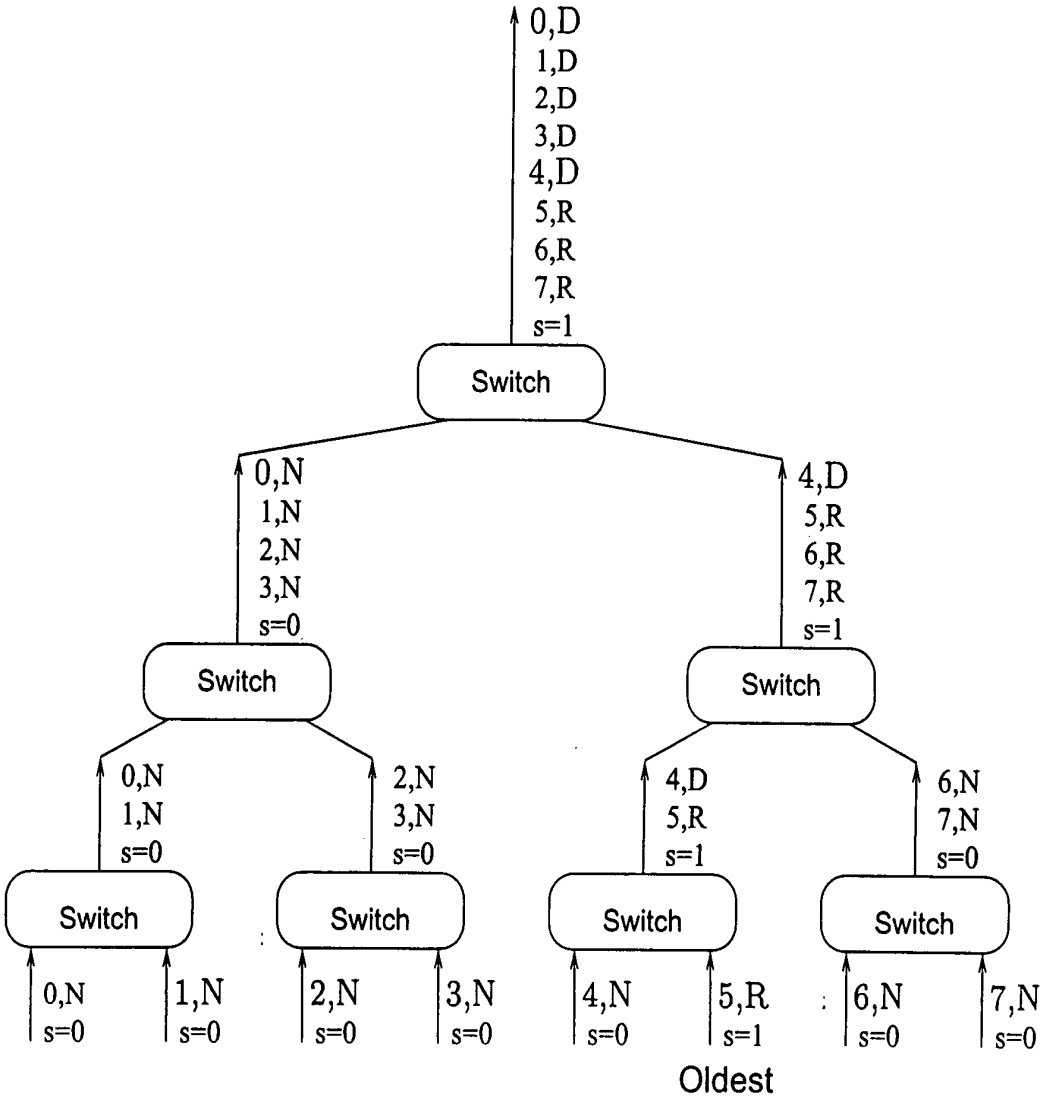

0,D
1,D
2,D
3,D
4,D
5,R
6,R
7,R
s=1
Switch
0,N
1,N
2,N
3,N
s=0
4,D
5,R
6,R
7,R
s=1
Switch
Switch
0,N
1,N
s=0
2,N
3,N
s=0
4,D
5,R
s=1
6,N
7,N
s=0
Switch
Switch
Switch
Switch
0,N
s=0
1,N
s=0
2,N
s=0
3,N
s=0
4,N
s=0
5,R
s=1
6,N
s=0
7,N
s=0
Oldest

FIG. 58

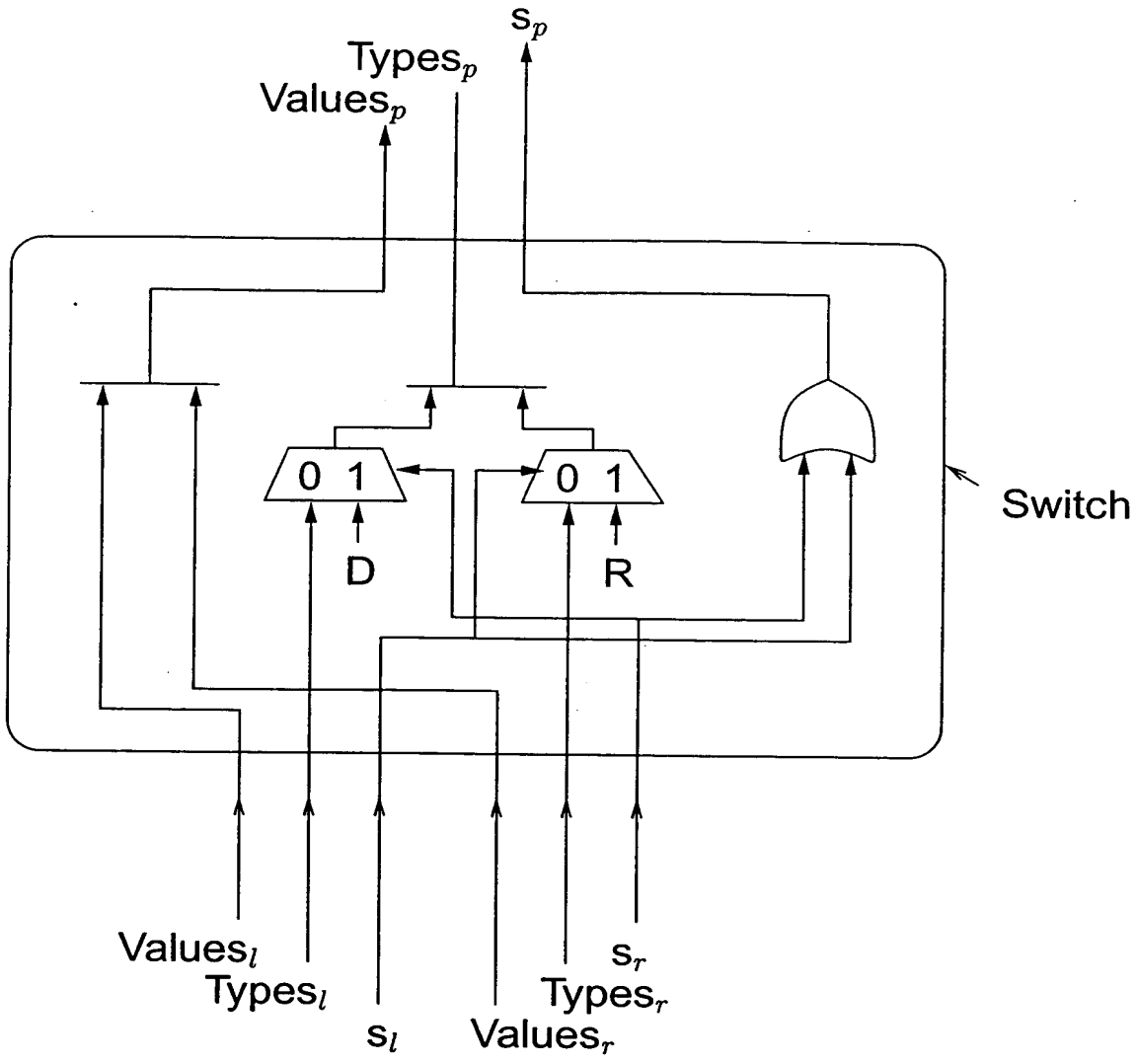

s_p
$Types_p$
$Values_p$
0 1
D
0 1
R
Switch
$Values_l$
$Types_l$
s_l
$Values_r$
s_r
$Types_r$

FIG. 59

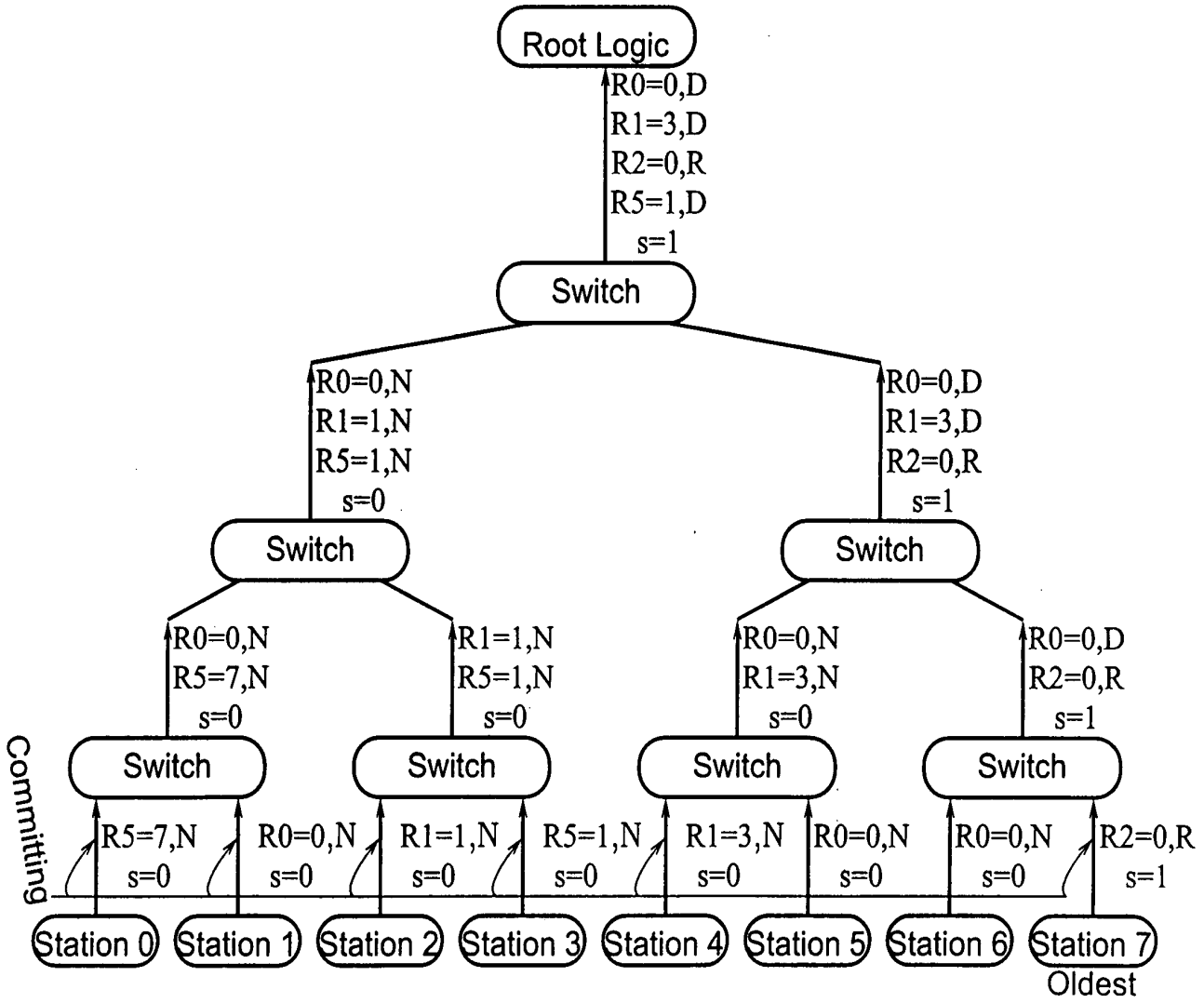

Root Logic
R0=0,D
R1=3,D
R2=0,R
R5=1,D
s=1
Switch
R0=0,N
R1=1,N
R5=1,N
s=0
R0=0,D
R1=3,D
R2=0,R
s=1
Switch
Switch
R0=0,N
R5=7,N
s=0
R1=1,N
R5=1,N
s=0
R0=0,N
R1=3,N
s=0
R0=0,D
R2=0,R
s=1
Switch
Switch
Switch
Switch
Committing
R5=7,N
s=0
R0=0,N
s=0
R1=1,N
s=0
R5=1,N
s=0
R1=3,N
s=0
R0=0,N
s=0
R0=0,N
s=0
R2=0,R
s=1
Station 0
Station 1
Station 2
Station 3
Station 4
Station 5
Station 6
Station 7
Oldest

FIG. 60

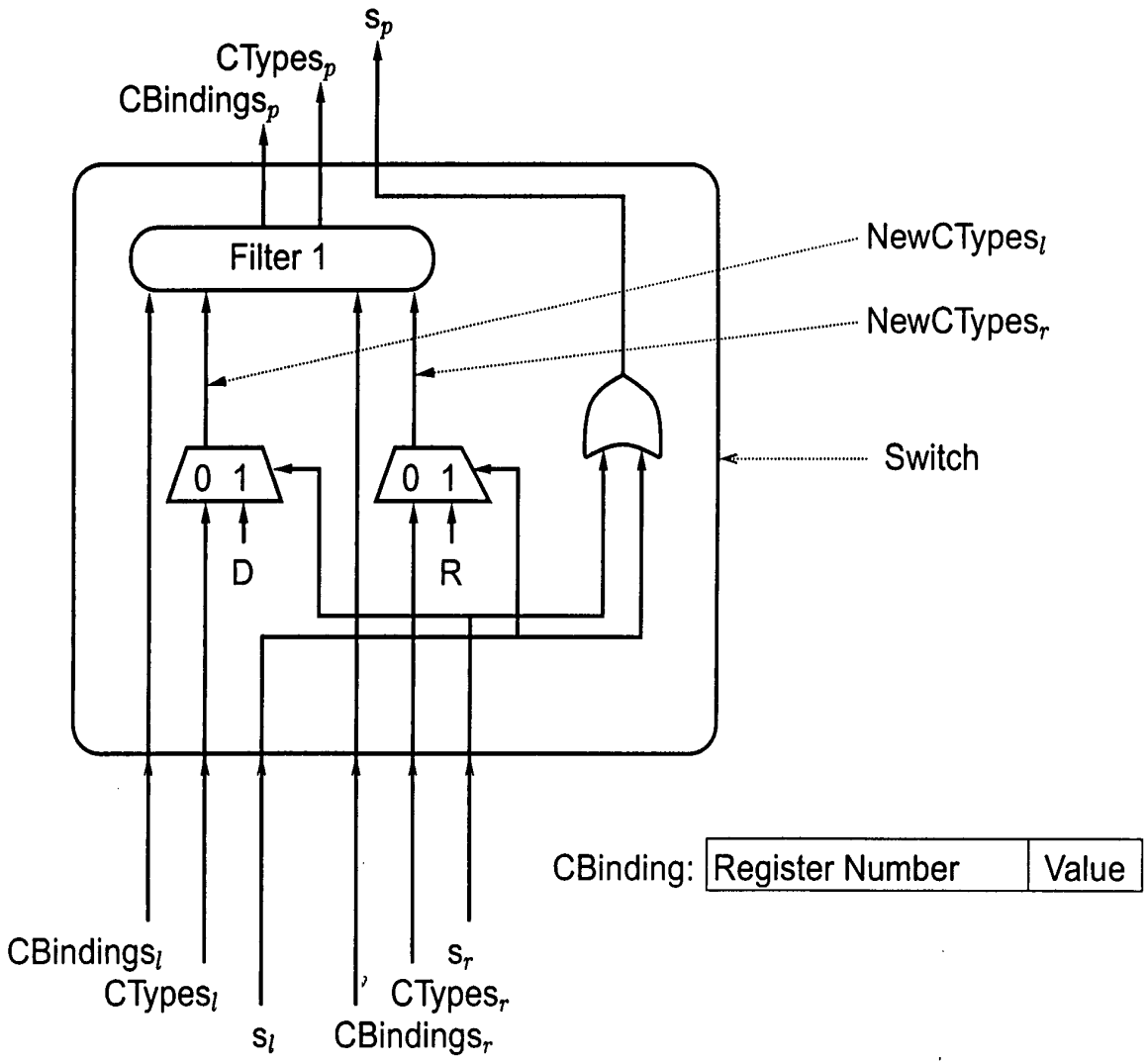

s_p
$CTypes_p$
$CBindings_p$
Filter 1
$NewCTypes_l$
$NewCTypes_r$
Switch
0 1
D
0 1
R
CBinding:
Register Number
Value
$CBindings_l$
$CTypes_l$
s_l
$CBindings_r$
$CTypes_r$
s_r

FIG. 61

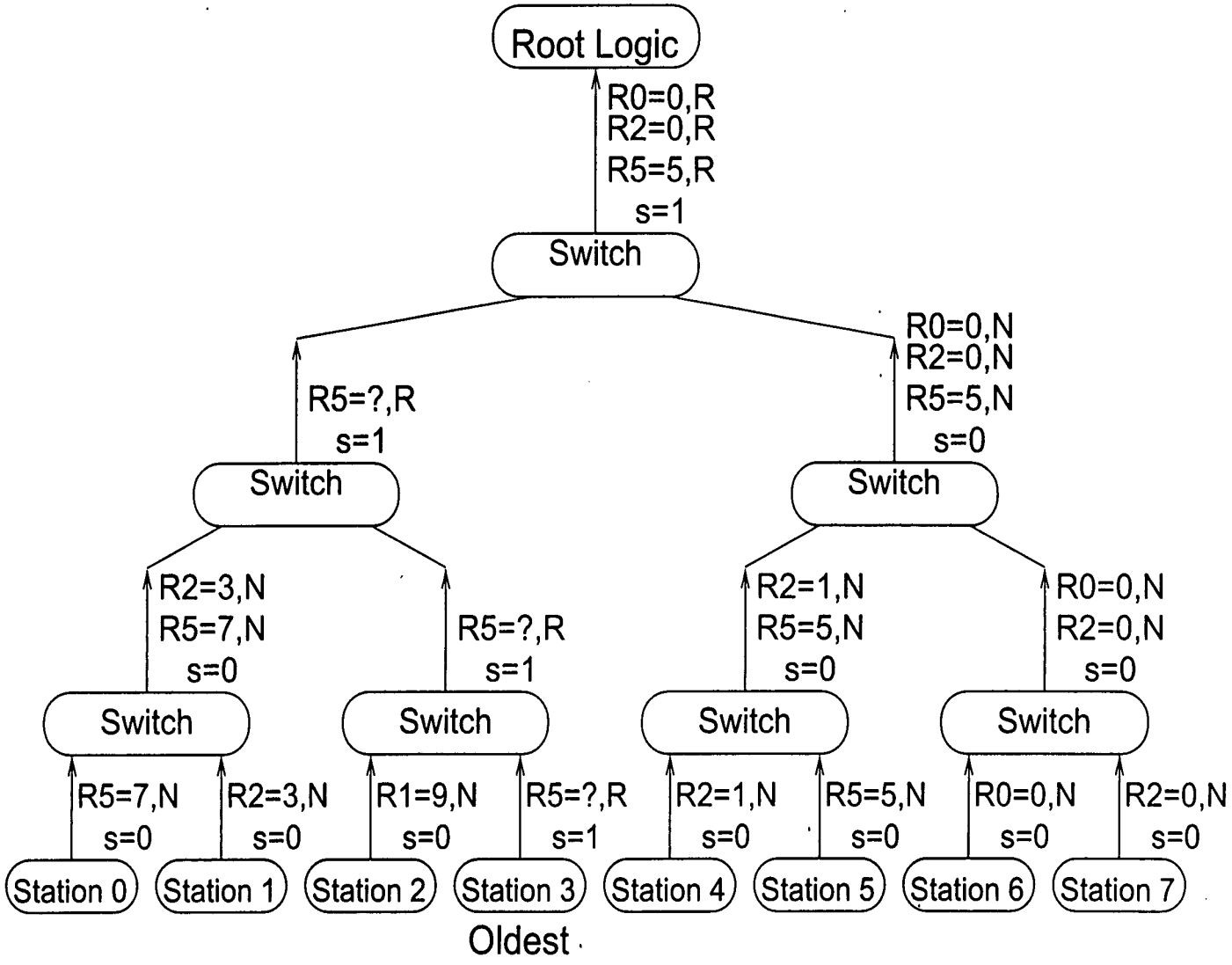

Root Logic
R0=0,R
R2=0,R
R5=5,R
s=1
Switch
R5=?,R
s=1
Switch
R0=0,N
R2=0,N
R5=5,N
s=0
Switch
R2=3,N
R5=7,N
s=0
Switch
R5=?,R
s=1
Switch
R2=1,N
R5=5,N
s=0
Switch
R0=0,N
R2=0,N
s=0
Switch
R5=7,N
s=0
R2=3,N
s=0
R1=9,N
s=0
R5=?,R
s=1
R2=1,N
s=0
R5=5,N
s=0
R0=0,N
s=0
R2=0,N
s=0
Station 0
Station 1
Station 2
Station 3
Station 4
Station 5
Station 6
Station 7
Oldest

FIG. 62

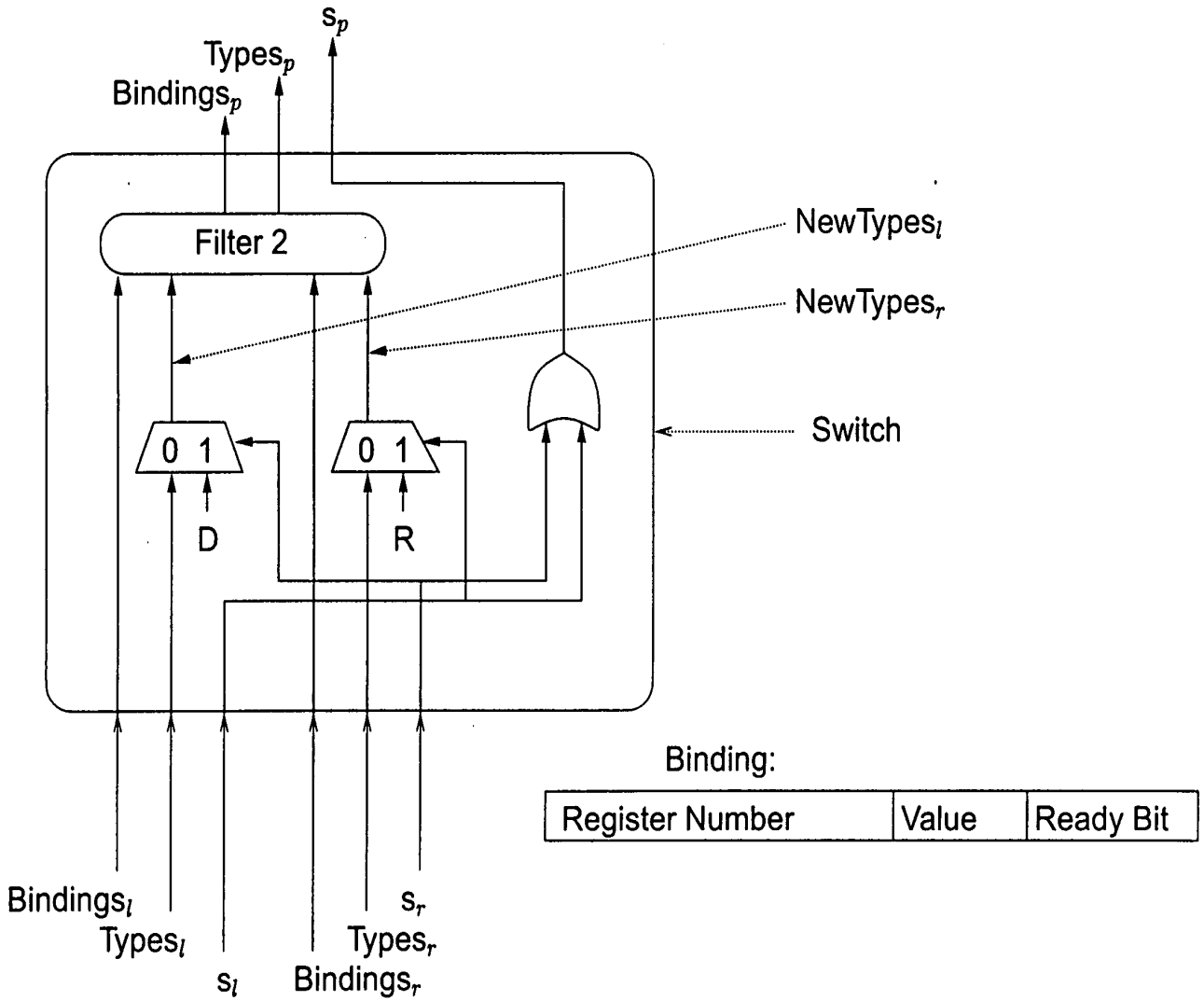

s_p
$Types_p$
$Bindings_p$
Filter 2
$NewTypes_l$
$NewTypes_r$
Switch
0 1
0 1
D
R
$Bindings_l$
$Types_l$
s_l
$Types_r$
$Bindings_r$
s_r
Binding:
Register Number
Value
Ready Bit

FIG. 63

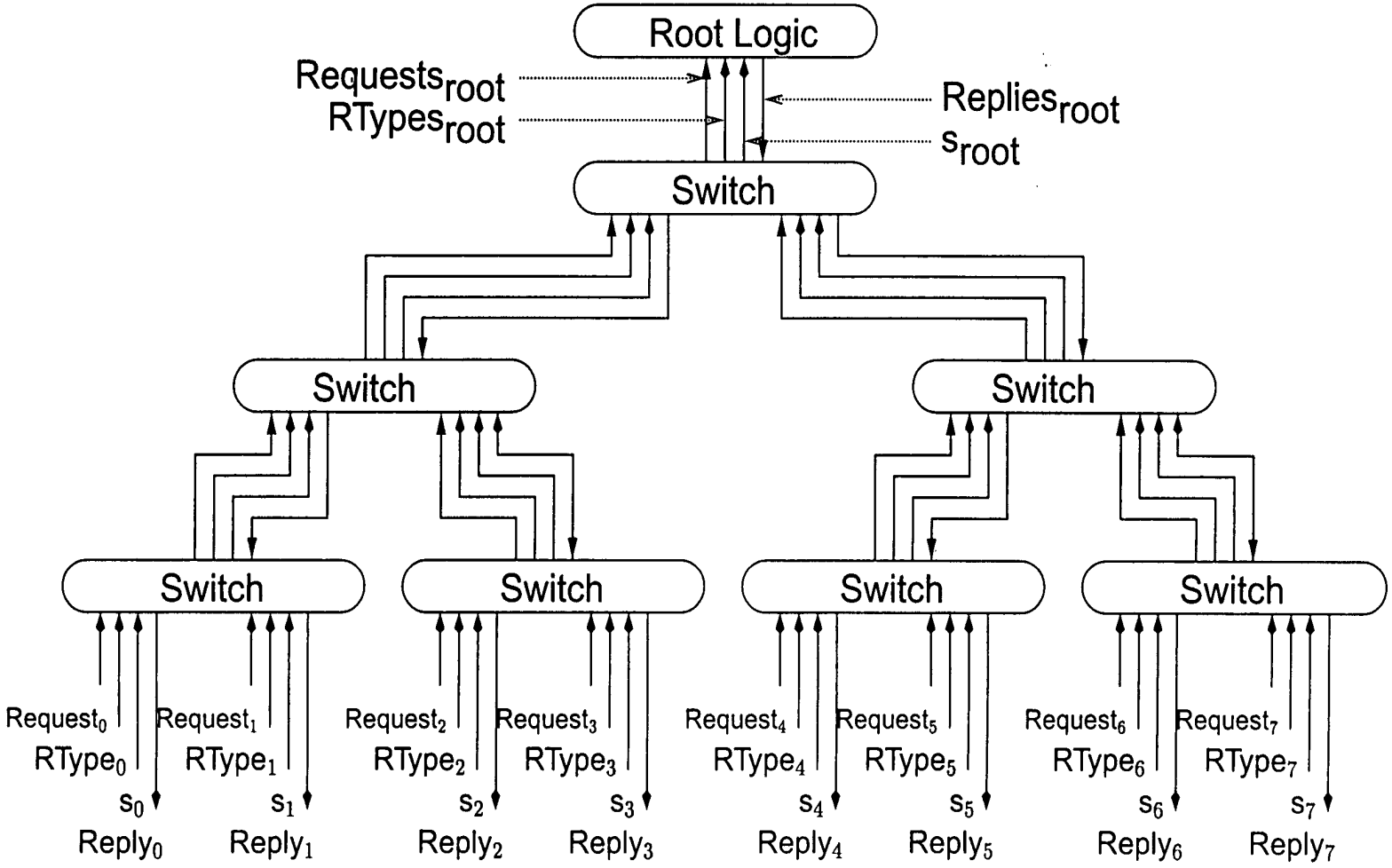

Root Logic
Requests$_{root}$
RTypes$_{root}$
Replies$_{root}$
s$_{root}$
Switch
Switch
Switch
Switch
Switch
Switch
Switch
Request$_0$
RType$_0$
s$_0$
Reply$_0$
Request$_1$
RType$_1$
s$_1$
Reply$_1$
Request$_2$
RType$_2$
s$_2$
Reply$_2$
Request$_3$
RType$_3$
s$_3$
Reply$_3$
Request$_4$
RType$_4$
s$_4$
Reply$_4$
Request$_5$
RType$_5$
s$_5$
Reply$_5$
Request$_6$
RType$_6$
s$_6$
Reply$_6$
Request$_7$
RType$_7$
s$_7$
Reply$_7$

FIG. 64

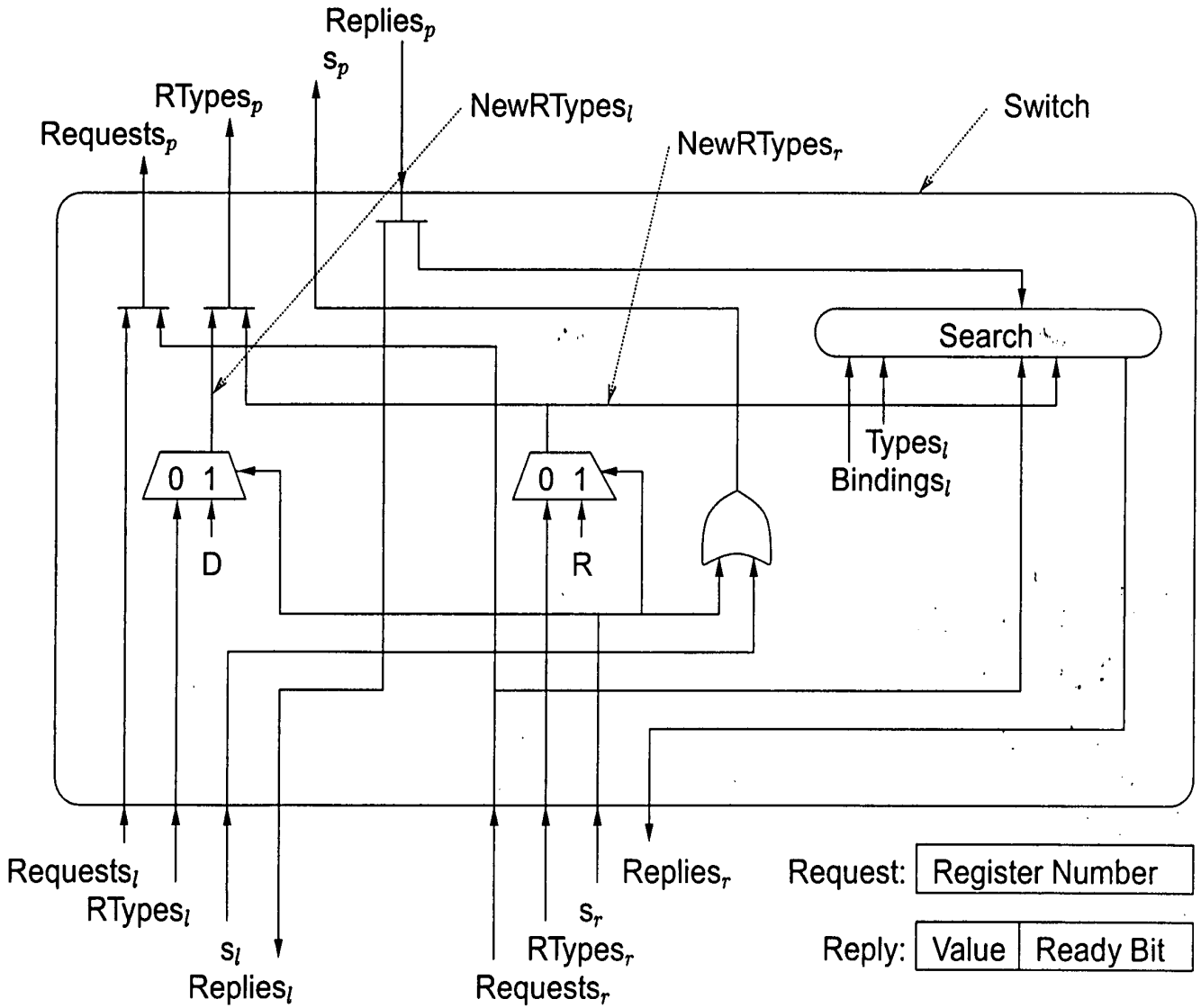

Replies$_p$
s_p
RTypes$_p$
Requests$_p$
NewRTypes$_l$
NewRTypes$_r$
Switch
Search
Types$_l$
Bindings$_l$
0 1
D
0 1
R
Requests$_l$
RTypes$_l$
s_l
Replies$_l$
Requests$_r$
RTypes$_r$
s_r
Replies$_r$
Request: Register Number
Reply: Value | Ready Bit

FIG. 65